



**A MULTI-OBJECTIVE, DECOMPOSITION-BASED
ALGORITHM DESIGN METHODOLOGY
AND ITS APPLICATION
TO RUNWAY OPERATIONS PLANNING**

Ioannis D. Anagnostakis and John-Paul Clarke

Report No. ICAT-2004-5
September 2004

***MIT International Center for Air Transportation
Department of Aeronautics & Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139, USA***

**A MULTI-OBJECTIVE, DECOMPOSITION-BASED
ALGORITHM DESIGN METHODOLOGY
AND ITS APPLICATION
TO RUNWAY OPERATIONS PLANNING**

BY

IOANNIS D. ANAGNOSTAKIS AND JOHN-PAUL CLARKE

ABSTRACT

Significant delays and resulting environmental impacts are commonly observed during departure operations at major US and European airports. One approach for mitigating airport congestion and delays is to exercise tactical operations planning and control with an objective to improve the efficiency of surface and terminal area operations. As a subtask of planning airport surface operations, this thesis presents *a thorough study of the structure and properties of the Runway Operations Planning (ROP) problem*. Runway Operations Planning is a workload-intensive task for controllers because airport operations involve many parameters, such as departure demand level and timing that are typically characterized by a highly dynamic behavior. This research work provides insight to the nature of this task, by analyzing the different parameters involved in it and illuminating how they interact with each other and how they affect the main functions in the problem of planning operations at the runway, such as departure runway throughput and runway queuing delays. Analysis of the Runway Operations Planning problem revealed that there is a parameter of the problem, namely the demand “weight class mix”, which: a) is more “dominant” on the problem performance functions than other parameters, b) changes value much slower than other parameters and c) its value is available earlier and with more certainty than the value of other parameters. These observations enabled the parsing of the set of functions and the set of parameters in subsets, so that the problem can be addressed sequentially in more than one stage where different parameter subsets are treated in different stages. Thus, *a decomposition-based algorithm design technique* was introduced and applied to the design of *a heuristic decomposed algorithm* for solving the ROP problem. This decomposition methodology offers an original paradigm potentially applicable to the design of solution algorithms for a class of problems with functions and parameters that, similar to those of the ROP problem, can be parsed in subsets. The potential merit in decomposing the ROP problem in two stages and the resulting utility of the two-stage solution algorithm are evaluated by performing *benefits analysis across specific dimensions* related to airport efficiency, as well as *stability and robustness analysis* of the algorithm output.

This document is based on the thesis of Ioannis D. Anagnostakis submitted to the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Aeronautics and Astronautics.

ACKNOWLEDGEMENTS

The ones I thank are family, friends, acquaintances and people that I met and possibly never spent too much time with but it was enough to detect valuable elements in their personalities that helped me to become who I am and be where I am today. To my long-time friend and academic advisor, Professor John-Paul Clarke, thank you for your guidance and support throughout my time at MIT. I owe you more than just a simple “a lot”. To Professors Amedeo Odoni, R. John Hansman and Peter Belobaba, thank you for inspiring me to work in the field of air traffic control and air transportation in general, since the time I was a Master’s student and for your guidance and advice throughout my research. To Prof. Emmy Roth, huge appreciation and admiration for personifying such a kind, patient and helpful contribution to my path through MIT. Special appreciation to Dr. Dietmar Böhme of the German Aerospace Research Establishment, Flight Guidance Division (DLR, Deutsches Zentrum für Luft- und Raumfahrt e.V., Institut für Flugführung) in Braunschweig, Germany, where I spend a very rewarding summer of year 2000.

I must acknowledge the help and contribution of all the Air Traffic Control Tower personnel at Logan Airport, who were instrumental to this research. Also, many thanks to all the colleagues who helped in conducting field observations and collecting data at the Logan Control Tower. Special thanks to the NASA Ames Research Center for supporting our research and more specifically to Dr. Leonard Tobias and Dr. Steve Atkins.

Many thanks and best wishes for the future to all the friends and colleagues in the International Center for Air Transportation at MIT and above all to: a) Dr. Husni Idris, whom I have known and admired for a long time, and b) Dr. Michael (Mikey) Clarke, a friend for more than a decade and a colleague in air transportation. I hope I can return the favors one day my friends.

To all my friends in the USA, Greece, Japan and all over the world, thanks for your caring, support and encouragement, you all made a difference in my life even from a distance. To my two best friends in the world: Dr. Dimitrios Kyritsis (?? ? ?μ?t??? ? ??λs??) and Dr. Spyridoula (Polly) Preventza (?? Sp??d??a ? ?eß?t?a). I am glad to have you in my life ...

To my father Dimosthenis Anagnostakis (? ?μ?s????? ? ?a??? st????) and to my mother Akrivi Anagnostaki Liarri (??B? ? ?a??? st??? ? ???). Being away from them and missing them made me feel even more how incredible people they are. I love you and dedicate this work to you. I just hope that I will be able to make my kids say and feel about me one day whatever I am feeling for you. To my brother Christos Anagnostakis (??st?? ? ?a??? st????) who is a brilliant man and is always in my mind and my heart.

To all those who really made me feel how much they love me or loved me (if you wish), even

though I might have not been ready to accept their love in some cases. If you ever read these paragraphs, you know who you are.

To God (whatever that concept may mean for each of you out there) for giving me the health and the strength to go on and struggle and complete this work. I hope that it will contribute something to this world. I also sincerely hope that my way through MIT and my efforts to obtain this degree did not make anyone hate me or think badly of me. I apologize if I did and I promise to make it up to you wherever and whoever you are.

What I appreciate most from all of the above people is that, in one way or another, they taught me what is the best way to live life: be kind to others, honest, courageous, loving, caring, understanding, generous and above all ... forgiving. Of course, it is another story that it took me several years and several thousands of beers to learn all the above from them ☺☺☺☺

*St?? ?????e? μ??:
St?? pat??a μ?? ? μ?s???? ? a??? st???
St?? μ?t??a μ?? ???β? ? a??? st??? ????
St?? ade?f? μ?? ???st? ? a??? st???*

*To my family:
My father Dimosthenis Anagnostakis
My mother Akrivi Anagnostaki Liarri
My brother Christos Anagnostakis*

Table Of Contents

CHAPTER 1 INTRODUCTION.....	29
1.1 PROBLEM STATEMENT	31
1.2 MAIN RESEARCH PREMISE.....	33
1.3 RESEARCH APPROACH	35
1.4 OVERVIEW OF THE DOCUMENT.....	36
CHAPTER 2 BACKGROUND.....	39
2.1 SURFACE OPERATIONS MANAGEMENT (SOM).....	39
2.1.1 <i>Literature Review: Surface Operations Management (SOM)</i>	40
2.2 SURFACE OPERATIONS PLAN GENERATION	44
2.2.1 <i>ROP in the Context of Air Traffic Management (ATM)</i>	46
2.2.2 <i>Literature Review: Runway Operations Planning (ROP)</i>	48
2.3 RESEARCH APPROACH	51
CHAPTER 3 DECOMPOSITION-BASED ALGORITHM DESIGN & THE RUNWAY OPERATIONS PLANNING PROBLEM.....	55
3.1 RUNWAY USAGE.....	56
3.2 DECOMPOSITION-BASED DESIGN OF THE ROP ALGORITHM	59
3.2.1 <i>Overview of the ROP Algorithm</i>	60
3.2.2 <i>Characteristics of the ROP Algorithm</i>	62
3.3 THEORETICAL BACKGROUND	66
3.3.1 <i>Literature Review: Decomposition in Design</i>	66
3.3.2 <i>The Design Process</i>	74
3.4 THEORY: OPERATIONS PLANNING PROBLEM.....	77
3.4.1 <i>Functional Requirements of the Operations Planning Problem</i>	79
3.4.2 <i>Design Parameters of the Operations Planning Problem</i>	80
3.4.3 <i>Decomposition of the Operations Planning Problem</i>	81
3.5 APPLICATION: RUNWAY OPERATIONS PLANNING PROBLEM	89

Table of Contents

3.5.1	<i>Decomposition Formulation</i>	89
3.5.2	<i>Additional Decomposition Dimensions: Degree of Impact</i>	97
3.5.3	<i>Additional Decomposition Dimensions: Time scale and Information Certainty</i>	99
3.6	TWO-STAGE SOLUTION APPROACH.....	106
3.7	STAGE ONE OF THE SOLUTION METHODOLOGY	112
3.7.1	<i>Constraints</i>	112
3.7.2	<i>Objective Function</i>	114
3.7.3	<i>The Departure Class Sequencer</i>	115
3.7.4	<i>Output</i>	116
3.8	STAGE TWO OF THE SOLUTION METHODOLOGY	116
3.8.1	<i>Constraints</i>	117
3.8.2	<i>Objective Function</i>	123
3.8.3	<i>The Departure Aircraft Scheduler</i>	126
3.8.4	<i>Output</i>	127
CHAPTER 4 METHODOLOGY FOR GENERATING GATE PUSHBACK PLANS FROM RUNWAY OPERATIONS PLANS.....		129
4.1	THE SIGNIFICANCE OF SURFACE MOVEMENT UNCERTAINTY IN RUNWAY OPERATIONS PLANNING.....	130
4.2	DETERMINING THE APPROPRIATE BACK-PROPAGATION ADJUSTMENT	134
CHAPTER 5 SIMULATION TEST BED.....		145
5.1	BENCHMARK AIRPORT	146
5.2	OVERVIEW OF THE SIMULATION MODEL	147
5.3	MODEL COMPONENTS	148
5.3.1	<i>“Gate to Runway” Aircraft Propagation Times</i>	148
5.3.2	<i>Generating Crossing Requests</i>	151
5.3.3	<i>Generating Possible Class Sequences</i>	151
5.3.4	<i>Mixing Departures and Crossings</i>	154
5.4	SIMULATION OF THE FIRST STAGE OF THE ALGORITHM	155
5.4.1	<i>Generating Class Sequences</i>	155

Table of Contents

5.4.2	<i>Introducing Crossings in the Class Sequences.....</i>	155
5.4.3	<i>Calculating the Throughput of Class Sequences.....</i>	156
5.5	SIMULATION OF THE SECOND STAGE OF THE ALGORITHM.....	160
5.5.1	<i>Entering Air Traffic Control Flow Restrictions.....</i>	160
5.5.2	<i>Branch & Bound Solution.....</i>	161
5.5.3	<i>2nd Stage Problem Size and Simulation Runtime.....</i>	162
5.6	SOLUTION EXAMPLES.....	163
5.6.1	<i>Stage One.....</i>	164
5.6.2	<i>Stage Two.....</i>	172
CHAPTER 6 SIMULATION TESTS & RESULTS.....		175
6.1	DEMAND TRAFFIC SCENARIOS.....	175
6.2	EVALUATION OF ALGORITHM PERFORMANCE.....	177
6.3	BENEFITS ANALYSIS: A REPRESENTATIVE EXAMPLE CASE - BOSTON LOGAN AIRPORT.....	178
6.3.1	<i>Airport Operational Performance Metrics.....</i>	180
6.3.2	<i>Methodology for Evaluating Planning Benefits.....</i>	181
6.3.3	<i>Potential Benefits from Planning in the Absence of Air Traffic Flow Restrictions.....</i>	190
6.3.4	<i>The Effect of Air Traffic Flow Restrictions on Potential Planning Benefits.....</i>	194
6.4	THE EFFECT OF PUSHBACK TIME UNCERTAINTY ON THE ROBUSTNESS OF RUNWAY OPERATIONS PLANS.....	213
6.4.1	<i>Methodology for Testing Robustness of Runway Operations Plans.....</i>	215
6.4.2	<i>Robustness Results.....</i>	220
6.4.3	<i>The Effect of ADAPTIVE Planning.....</i>	223
6.5	THE EFFECT OF PUSHBACK TIME UNCERTAINTY ON THE STABILITY OF RUNWAY OPERATIONS PLANS.....	225
6.5.1	<i>Methodology.....</i>	226
6.5.2	<i>Stability Metrics.....</i>	227
6.5.3	<i>Stability with Respect to Weight Class and Aircraft Sequences.....</i>	227
6.6	THE EFFECT OF PUSHBACK TIMING.....	233
6.6.1	<i>Scenarios WITHOUT Planning.....</i>	236
6.6.2	<i>Scenarios WITH Planning.....</i>	239

6.7	THE EFFECT OF DEMAND WEIGHT CLASS MIX.....	242
6.7.1	<i>Scenarios WITHOUT Planning</i>	243
6.7.2	<i>Scenarios WITH Planning</i>	246
6.7.3	<i>Stability Analysis</i>	247
6.8	PUSHBACK POSITION SHIFTS VS. TAKEOFF POSITION SHIFTS.....	248
6.9	ROBUSTNESS AND STABILITY IN THE PRESENCE OF AIR TRAFFIC CONTROL RESTRICTIONS.....	251
CHAPTER 7 CONCLUSIONS & IMPLICATIONS.....		253
7.1	SUMMARY OF RESEARCH CONTRIBUTIONS.....	254
7.1.1	<i>Decomposing the Runway Operations Planning Problem</i>	254
7.1.2	<i>Runway Operations Planning (ROP) Algorithm</i>	258
7.1.3	<i>Using the runway operations plans suggested by the ROP algorithm</i>	259
7.2	SUMMARY OF RESEARCH FINDINGS.....	261
7.2.1	<i>Effect of Active Flow Restrictions</i>	262
7.2.2	<i>Effect of Pushback Timing</i>	264
7.2.3	<i>Effect of the Weight Class Mix</i>	265
7.2.4	<i>Effect of Pushback Uncertainty</i>	267
7.3	IMPLICATIONS FOR REAL-WORLD PROBLEMS.....	269
7.4	FUTURE RESEARCH.....	271
REFERENCES		275
APPENDIX A FIELD OBSERVATIONS.....		289
A.1	THE DEPARTURE PROCESS –FIELD OBSERVATIONS.....	289
A.2	OPERATIONAL INEFFICIENCIES OBSERVED.....	296
APPENDIX B CONCEPTUAL ARCHITECTURE OF A DEPARTURE PLANNING DECISION-AIDING SYSTEM.....		301
B.1	CONFIGURATION PLANNER.....	307
B.2	THE VIRTUAL QUEUE MANAGER.....	311
B.3	THE GATE MANAGER.....	318
B.4	THE TAXIWAY ENTRY MANAGER.....	322

B.5	THE MIX MANAGER	326
APPENDIX C	RUNWAY OPERATIONS PLANNING - FUNCTIONAL AND RELATIONAL DECOMPOSITION ANALYSIS	331
C.1	FUNCTIONAL DECOMPOSITION ANALYSIS	332
C.1.1	ROP “Customer” Needs & Functional Requirements	332
C.1.2	ROP Design Parameters & Process Variables.....	336
C.2	ROP PROBLEM OBJECTIVES & CONSTRAINTS.....	338
C.2.1	Constraint Types	338
C.2.2	Constraint Categories.....	340
C.3	RELATIONAL DECOMPOSITION ANALYSIS	344
C.3.1	Separate Aircraft (Safety).....	345
C.3.2	Maximize Throughput.....	346
C.3.3	Minimize Delay.....	349
C.3.4	Be Fair to All Users	350
C.3.5	Satisfy Downstream Air Traffic Flow Restrictions	351
APPENDIX D	SIMULATION OUTPUT	353
D.1	1 ST STAGE OUTPUT: WEIGHT CLASS SEQUENCES	353
D.1.1	Introducing Crossings in the Runway Operations Plan.....	357
D.2	2 ND STAGE OUTPUT: RUNWAY OPERATIONS PLANS	361
D.2.1	Output Format.....	361
D.2.2	Allocation of Aircraft to Weight Class Slots.....	364
APPENDIX E	CONVERGENCE OF THE MONTE CARLO SIMULATION	371
APPENDIX F	STANDARD ERROR ANALYSIS.....	375
F.1	EFFECT OF AIR TRAFFIC FLOW RESTRICTIONS	375
F.2	EFFECT OF PUSHBACK TIMING.....	376
F.3	EFFECT OF DEMAND WEIGHT CLASS MIX	377
APPENDIX G	RUNWAY OPERATIONS PLANNING IN ONE STAGE.....	379

List of Figures

Figure 1.1: Domestic traffic growth (Bureau of Transportation Statistics data)	30
Figure 2.1: Surface Operations Management: task structure	40
Figure 2.2: Surface Operations Management in the context of Air Traffic Management	47
Figure 3.1: Layout of Airport Cologne-Bonn (sketch).....	57
Figure 3.2: Runway Operations Planning in two stages.....	61
Figure 3.3: Four Domains of the Design World (source [116]).....	76
Figure 3.4: Design Matrix equation (example) - Mapping between Functional Requirements {FR} & Design Parameters {DP}	82
Figure 3.5: Prediction Accuracy of Departure Demand in terms of Traffic Mix vs. Tail Number	100
Figure 3.6: % Difference between Predicted and Actual % HEAVIES Demand (BOS, Feb. 2, 1999).....	103
Figure 3.7: % Difference between Predicted and Actual % LARGE Demand (BOS, Feb. 2, 1999)	103
Figure 3.8: Flight Specific Pushback Time Prediction Error: Flight Without Delay	104
Figure 3.9: Flight Specific Pushback Time Prediction Error: Flight With Delay	104
Figure 3.10: ROP – Mapping between Functional Requirements {FR} & Design Parameters {DP}	110
Figure 3.11: Design Quality & Design Matrix.....	110
Figure 4.1: Example of Reducing the Likelihood of a “Runway-Idle” Gap by Adjusting Earlier the Planned Pushback Time of the Trailing Aircraft.....	132
Figure 4.2: Analyzing the Trend of the Differential in Pushback Time “Adjustment”.....	136
Figure 4.3: Suggested Quadratic Trend for the Pushback Time “Adjustment”.....	138
Figure 4.4: Logan Configuration 22L / 22R / 27 – Queuing Time Behavior	140
Figure 5.1: Benchmark example airport system	146
Figure 5.2: Simulink Model Overview.....	148
Figure 5.3: Propagator	149
Figure 5.4: Possible class slot swaps	158
Figure 5.5: Sequences derived from the “base” sequence by performing all possible swaps.....	158
Figure 5.6: Probability of Occurrence of a Slot Swap.....	159
Figure 5.7: GUI for In Trail restrictions	160
Figure 5.8: GUI for Time Window restrictions	160
Figure 6.1: The “Planner” - Planning Gate Pushback Operations by Using the Results of the Runway Operations Planning Algorithm	182
Figure 6.2: Using the “Planner” and the FCFS simulation for evaluating different queuing curves for the back-propagation queuing adjustment of the Planner (Figure 6.1)	184
Figure 6.3: Final vs. Original Queuing Curve.....	186
Figure 6.4: Comparing Airport Operational Performance between Scenarios WITH and WITHOUT Planning	188
Figure 6.5: Overall Benefits Analysis Methodology - Comparing Airport Operational Performance between Scenarios WITH and WITHOUT Planning under a FCFS Ground Traffic Handling Strategy.....	189
Figure 6.6: Conceptual Architecture of the Departure Planner – Simplified Version.....	190

Figure 6.7: Departure and Total Throughput	191
Figure 6.8: Cumulative Departure Delay and Elapsed Time	192
Figure 6.9: “Aircraft to Slot” assignments in an Unrestricted vs. a Restricted Traffic Scenario	196
Figure 6.10: “Aircraft to slot” assignments with a First Come First Serve Aircraft Handling Strategy Under Miles In Trail Restrictions.....	199
Figure 6.11: Departure Throughput.....	204
Figure 6.12: Total Throughput.....	204
Figure 6.13: Cumulative Departure Delay	204
Figure 6.14: Cumulative Elapsed Time.....	204
Figure 6.15: Departure Throughput.....	209
Figure 6.16: Total Throughput.....	209
Figure 6.17: Cumulative Departure Delay	210
Figure 6.18: Cumulative Elapsed Time.....	210
Figure 6.19: “Aircraft to slot” assignments with a First Come First Serve Aircraft Handling Strategy Under Miles In Trail and DSP Restrictions.....	212
Figure 6.20: Robustness Analysis WITHOUT Planning	218
Figure 6.21: Robustness Analysis WITH Planning.....	218
Figure 6.22: Robustness Analysis WITH ADAPTIVE Planning	219
Figure 6.23: Robustness of Departure Throughput WITH Planning.....	220
Figure 6.24: Robustness of Departure Delays WITH Planning.....	221
Figure 6.25: Departure Throughput Robustness Analysis	222
Figure 6.26: Departure Delays Robustness Analysis.....	223
Figure 6.27: Stability of Algorithm Solutions - Pushback Uncertainty Standard Deviation of 180 sec	228
Figure 6.28: Stability of Algorithm Solutions when One (1) Aircraft has a Changed Pushback Time.....	231
Figure 6.29: Stability of Algorithm Solutions when Two (2) Aircraft have Changed Pushback Times	231
Figure 6.30: Pushback Schedules with Different Pushback Time Distributions	234
Figure 6.31: Effect of Pushback Timing on Departure Throughput.....	236
Figure 6.32: Effect of Pushback Timing on Departure Delay	238
Figure 6.33: Effect of Pushback Timing on Elapsed Time	238
Figure 6.34: Effect of Schedule Weight Class Mix on Departure Throughput	244
Figure 6.35: Effect of Schedule Weight Class Mix on Departure Delay	245
Figure 6.36: Effect of Schedule Weight Class Mix on Elapsed Time.....	245
Figure 6.37: Stability of Algorithm Solutions - Pushback Uncertainty Standard Deviation of 120 sec, Balanced Weight Class Mix (Heavy, Large and Small, 33.33 % each) in the Original Pushback Schedule	247
Figure 6.38: Change in Algorithm Input vs. Algorithm Output (All 500 Runs)	249
Figure 6.39: Change in Algorithm Input vs. Algorithm Output (Only Runs with Optimal Class Sequence)	250
Figure 7.1: Design Matrix Equation – Flow Restrictions are Used as a Decomposition Pivot Element.....	263
Figure A.1: Taxiway and Runway Queues at Logan Airport under configuration 22L / 22R / 27	291

Figure A.2: Queuing Model for Logan Airport under configuration 22L / 22R / 27 (source [74])	291
Figure A.3: Miles In Trail (MIT) restrictions	297
Figure A.4: Takeoff sequence adjustments	299
Figure B.1: Conceptual Architecture for the Departure Planner	302
Figure B.2: The Configuration Planner	308
Figure B.3: Runway Configuration Capacity Envelopes (Graph source: [74], Data source: ETMS / Tower Records, 7-9 AM, 4-8 PM, July 1-15 1998 except Saturdays, Logan Airport).....	309
Figure B.4: Accelerated Departure Procedure (ADP) (Graph source: [74], Data source: ETMS, July 1-15 1998, Logan Airport).....	309
Figure B.5: Managing the departure sequence of the same 18 flights, with and without the implementation of a Virtual Queue (right hand side indicates the current optimal sequence)	314
Figure B.6: ROP implementation – “Expected” aircraft ARE included in the planning	317
Figure B.7: ROP implementation – “Expected” aircraft ARE NOT included in the planning...	317
Figure B.8: The Gate Manager.....	318
Figure B.9: Taxi out time as a function of airport congestion (BOS: 1997/09 – 1998/08).....	321
Figure B.10: Takeoff Rate saturation point (BOS: 1997/09 – 1998/08).....	321
Figure B.11: The Taxiway Entry Manager	323
Figure B.12: Runway Operations Schedule: Mixing of Arrivals, Departure and Runway Crossings (source: [74])	327
Figure B.13: The Mix Manager	328
Figure B.14: Runway Crossing Effect on Inter-Takeoff Clearance Times (Graph source: [74], Data source: Local Controller Clearances, Runway 22R, Logan Airport, 12-2-1998, 4-9PM)	329
Figure C.1: Layers of “customer” needs / attributes in Runway Operations Management	335
Figure C.2: ROP Design Matrix Equation – Step 1.....	345
Figure C.3: ROP Design Matrix Equation – Step 2.....	348
Figure C.4: ROP Design Matrix Equation – Final.....	351
Figure D.1: Departure Class Schedules with a Heavy as the “extra” aircraft	354
Figure E.1: Simulation Convergence Analysis – Stability Metrics	372
Figure E.2: Simulation Convergence Analysis – Departure Throughput.....	373
Figure E.3: Simulation Convergence Analysis – Departure Delay & Elapsed Time.....	373

List of Tables

Table 3.1: Interdependency matrix for Airport Cologne-Bonn in configuration 14L, 14R and 25.	57
Table 4.1: Disjoint Events that can lead to Slot 3 Being Covered by a Preceding Aircraft.....	137
Table 5.1: Example “aircraft to class slot” assignments	173
Table 6.1: Performance Comparison between Using the Original and the Final Queuing Curve in Planning.....	187
Table 6.2: Benefits Analysis for Traffic Scenarios WITHOUT Flow Restrictions (Earlier than Scheduled Pushback Times ARE Allowed).....	192
Table 6.3: Benefits Analysis for Traffic Scenarios WITHOUT Flow Restrictions (Earlier than Scheduled Pushback Times ARE NOT Allowed).....	194
Table 6.4: Example - “Aircraft to slot” assignments under Miles In Trail restrictions.....	195
Table 6.5: Benefits Analysis for a Restricted Traffic Scenario - EWR & JFK are restricted airports.....	202
Table 6.6: Estimated Annual Savings in Aircraft Fuel Burned and Reduction in Aircraft Engine Emissions (only BOS departures and only at periods of high traffic volume).....	207
Table C.1: ROP “Customer” Needs and Functional Requirements.....	335
Table C.2: Separation criteria (in sec) for three weight classes (Heavy, Large and Small).....	337
Table C.3: The Complete Set of ROP Design Domains	337
Table D.1: Example Output of the Two-Stage Algorithm (Runway Operations Plan).....	362
Table D.2: Example - Ground Traffic Performance Metrics and “Aircraft to Slot” Assignments under the same pushback schedule input and with respect to different 2 nd stage objective functions (no air traffic flow restrictions active)	368
Table F.1: Standard Error of the Mean Values of Performance Metrics for scenarios with and without Planning under Restricted and Unrestricted Traffic Conditions (re: Figure 6.11, Figure 6.12, Figure 6.13, and Figure 6.14).....	375
Table F.2: Standard Error of the Mean Values of Performance Metrics for a Restricted Scenario with and without Planning (re: Figure 6.15, Figure 6.16, Figure 6.17 and Figure 6.18)....	376
Table F.3: Standard Error of the Mean Values of Performance Metrics for Schedules with Varying Pushback Time Distributions (re: Figure 6.31, Figure 6.32 and Figure 6.33).....	376
Table F.4: Standard Error of the Mean Values of Performance Metrics for Schedules with Varying Pushback Time Distributions (re: Figure 6.31, Figure 6.32 and Figure 6.33).....	376
Table F.5: Standard Error of the Mean Values of Performance Metrics for Schedules with Evenly-Spaced Pushback Times (Regular) and Varying Demand Weight Class Mix (re: Figure 6.34, Figure 6.35 and Figure 6.36).....	377

List of Acronyms

A-SMGCS	Advanced Surface Movement Guidance and Control Systems
ASMS	Airport Surface Management System
ASP	Aircraft Sequencing Problem
ATC	Air Traffic Control
ATM	Air Traffic Management
CENA	Centre D' Etudes De La Navigation Aérienne (French Air Navigation Study Center)
CTAS	Center TRACON Automation System
DEPARTS	Departure Enhanced Planning And Runway/Taxiway-Assignment System
DLR	Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Research Establishment)
DMAN	Departure Manager
DP	Design Parameter
DSEDM	Departure Sequencing Engineering Design Model
DSP	Departure Sequencing Program
EDCT	Expected Departure Clearance Time
FAA	Federal Aviation Administration
FCFS	First Come First Serve
FR	Functional Requirement

MANTEA	Management of Surface Traffic in European Airports
MIT	Massachusetts Institute of Technology
MiIT	Miles In Trail
MinIT	Minutes in Trail
MPS	Maximum Position Shifting
ROM	Runway Operations Management
ROP	Runway Operations Planning
SMA	Surface Movement Advisor
SMS	Surface Management Systems
SOC	Surface Operations Control
SOM	Surface Operations Management
SOP	Surface Operations Planning
TARMAC	Taxi and Ramp Management and Control
TCS	Target Class Sequence
TGOP	Taxi and Gate/Ramp Operations Planning
TRACON	Terminal Radar Approach CONTROL

List of Symbols

ΔP_{ij}	Minimum required takeoff sequence position separation between flights i and j
ΔT_{ij}	Minimum time separation at the takeoff point between flights i and j
$EOff_i$	Earliest possible takeoff time of a departure operation
EOn_i	Earliest possible touchdown time of an arrival operation
EX_i	Earliest possible crossing time of a crossing operation
N	Total number of “mixed” operations on the runway (departures and arrivals)
N_A	Total number of Arrivals considered
N_D	Total number of Departures considered
N_S	Total number of slots in a class slot sequence
$P_{\max TO_i}$	Upper bound on the takeoff sequence position
P_{PB_i}	Pushback sequence position of aircraft i
P_{TO_i}	Takeoff sequence position of aircraft i
RT	Runway Throughput
$sDSP_i$	Takeoff slot boundary values for a DSP slot window
$sEDCT_i$	Takeoff slot boundary values for an EDCT slot window
t_{D_i}	Time of occurrence of a takeoff operation
$tDSP_i$	Time values that determine the start and end of a DSP time window

t_{EDCT_i}	Time values that determine the start and end of an EDCT time window
t_i	Time of occurrence of a runway operation (takeoff, landing or crossing)
T_i	Time of ADL prediction update (section 3.5.1)
$T_{m,n}$	Time length of the individual time gap between successive departure operations m and n
$TOff_i$	Actual takeoff time of a departure
TOn_i	Actual touchdown time of an arrival
T_{total}	Total time needed to complete a departure sequence
TX_i	Actual crossing time of a crossing operation
X_{ij}	Decision variables selected for the 2 nd stage problem formulation
WCM_i	Decision variable for the 1 st stage problem formulation
WCM_{PREDij}	“Prediction at time T_i ” value for the weight class mix expected to operate within time window j
WCM_{ACTij}	“Actual” value for the weight class mix of the aircraft group that actually pushed back from their gates within the boundaries of time window j

List of Simulation Inputs and Outputs

Inputs

Occ_H	Landing occupancy time (in sec) for a Heavy aircraft
Occ_L	Landing occupancy time (in sec) for a Large aircraft
Occ_S	Landing occupancy time (in sec) for a Small aircraft
DEPARR	Departures & Arrivals schedule (aircraft pool)
[start_ac,end_ac]	Aircraft window to be optimized
sch_limit	Number of iterations of the random generation process & class schedules generated
cross_cap_1, cross_cap_2	Cross point capacities
max_cross_delay	Maximum Crossing Delay
num_of_target_sequences	Number of Target Class Sequences to be considered

Outputs

XclassSchedule_stat	Array that contains all Target Class Sequences with crossings and a stochastic throughput value for each
Final_Schedule	Array that contains the selected Target Class Sequences and the aircraft timing for each of them
xfinal	2 nd stage solution, i.e. Final aircraft to slot assignments

List of Simulation Inputs and Outputs

tot_time	Total time to complete operations
tot_pos_shifts	Cumulative number of position shifts
TC	Total Cost, i.e. cumulative departure delay in the worst case (upper bound)

Chapter 1

INTRODUCTION

Prior to September 11, 2001, the demand for air traffic control services was increasing at a rate far in excess of the increases in airport and airspace capacity enabled by new Air Traffic Control (ATC) technologies (Figure 1.1). Almost 3 years after the fateful events of that day, passengers have returned to the skies as a result of a rebounding US economy and the lowest fares in a decade. Given this rebound in passenger demand, traffic is expected to surpass pre-September 11 levels. Consequently, a capacity / demand mismatch is still likely to impact the air transportation system (whether in the airspace between airports, adjacent to airports or on the airport surface). Such a mismatch is most frequently handled through ground holding and ground delay programs that keep departing aircraft on the ground due to capacity constraints at destination airports. Departures are also often delayed due to bad weather or high traffic volume in the terminal area or en-route. Thus, as the US Transportation Department recently reported, significant delays are still commonly observed during departure operations at major US airports [74]. Through May 2004:

- About 1 in 5 flights arrived more than 15 minutes behind schedule, which is the threshold for a flight to be considered delayed,
- About a quarter of all late flights arrived more than an hour behind schedule and
- The percentage of late arrivals increased by 27 % compared to the first five months of 2003, reversing any on-time performance improvements that had been recorded since 2000.

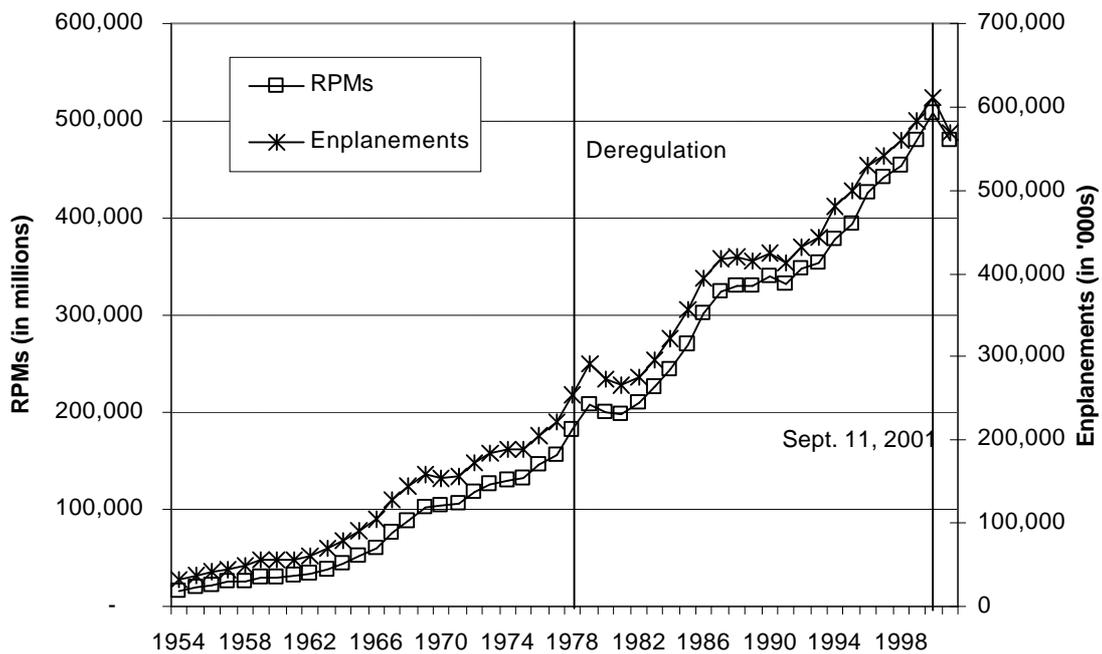


Figure 1.1: Domestic traffic growth (Bureau of Transportation Statistics data)

It is estimated that airport and airspace congestion now cost the US economy about 7 billion US dollars a year [26]. Such costs in the form of operational inefficiencies (e.g. fuel burn), as well as the environmental impact (primarily in the form of increased engine emissions) generate a growing need for the reduction of delays and mitigation of the inefficiencies on the airport surface and in the terminal airspace surrounding it.

1.1 Problem Statement

Due to the highly dynamic nature of air traffic control operations, airport ground operations are characterized by high uncertainty. More specifically, during departure operations, the aircraft system state, i.e. the aircraft (and other surface vehicle) position and intentions are highly unpredictable and only partially observable to controllers. Uncertainty in airport ground operations usually manifests on the airport surface in the form of congestion queues, especially in the absence of sophisticated decision-aiding tools for planning and controlling the departure flow. Surface congestion queues may result in significant operational inefficiencies, such as delays in the departure process, excessive fuel burn and considerable environmental impacts, such as aircraft emissions¹. Of all airport ground resources where inefficiencies manifest, field observations ([71], [72]) indicate that *the runway, which is shared by all types of operations (arrivals, departures and crossings), is a key airport resource and the main departure flow bottleneck.*

Furthermore, while air traffic controllers usually handle high volumes of airport traffic very well, observations also indicate that controllers have difficulty in situations where advanced planning and optimization of airport surface operations is required ([73], [74]). Particularly, air traffic controllers find it difficult to track and account for all the flow management constraints imposed on aircraft, which results in airport surface congestion and inefficiencies (long departure queues and unnecessary runway “idle” time).

¹ Other examples of such inefficiencies can be found in Appendix A.

Given the traffic growth expected at airports in the US, combined with the realities of daily airport operations mentioned above and the criticality of the runway as an airport resource, it seems that precise planning and execution of runway operations would enable efficient use of this critical resource and potentially could mitigate environmental impacts. Hence, *there is an apparent need to design and implement decision-aiding systems for airport surface operations* and particularly for runway operations planning and control, in order to:

1. Assist air traffic controllers in enhancing the performance of departure operations, by:

- Aiding them in tracking flow management constraints and
- Suggesting aircraft sequences and schedules that maximize throughput while satisfying all system constraints

and

2. Help in mitigating existing inefficiencies (reduce delays and associated costs and environmental impact), by:

- Exercising tighter sequencing and scheduling control on each portion of the departure process (in cooperation with existing arrival flow automation systems) and
- Controlling airport congestion (network of surface congestion queues)

To this purpose, various methods have been proposed to mitigate airport surface congestion and delays:

a) Capacity management by adding new runways and possibly new airports,

b) Demand management by applying peak pricing regulations on the available airport resources,

such as the gates and the runways, or

- c) Efficient management of the available airport resources by performing strategic and / or tactical operations planning and control to enhance the efficiency of surface and terminal area operations.

1.2 Main Research Premise

Because the airport runway was identified to be a primary constraint (“bottleneck”) for the departure flow [71], [72], planning of runway operations was chosen in this thesis as the first problem to be investigated in an effort to begin addressing the overall surface operations planning problem.

Until recently, a significant amount of Air Traffic Management (ATM) research on efficient management of surface and terminal area operations had been conducted. However, this research had been somewhat focused on the arrival flow [37], [96], [134] with departures typically only taken into account in an approximate manner. The reality is, as shown by previous research and field observations, that *arrivals and departures are highly coupled processes*, with complex interactions because they share many of the same airport and terminal airspace resources. So, given that any solution developed for only one group in a system with two groups is likely to be sub-optimal, it was realized that, only enhanced airport surface traffic control systems which fully include departures, could optimize the overall efficiency of airport operations [1], [22], [43], [53], [71], [72], [73], [126].

As it will be shown in the literature review of Chapter 2, in order to satisfy the need for such systems, there has been a significant amount of published research and *many different research schemes were proposed in the last few years on managing airport surface operations and more specifically, runway operations*. However, in several cases, the results have been characterized by limiting assumptions and most of the proposed schemes have included a significant amount of detail in their attempt to micro-manage every possible aspect of airport surface operations. In most cases, such a level of detail leads to systems that are difficult to implement due to:

- The dependence of system design on complicated algorithmic methods that lead to high computational overhead associated with the operation of such a system,
- The frequent assumption that sophisticated and accurate airport surface surveillance systems exist, even though, in many cases, the technology for such systems either does not exist or has not been implemented in a real-world operational environment yet and
- The human factors challenges that usually accompany such complicated systems that are destined to be operated and used by human operators, such as the air traffic controllers.

In this document, we propose a method for addressing the Runway Operations Planning (ROP) problem², which deviates from other suggested solution approaches and addresses the ROP problem to the level of actually scheduling runway operations on an “aircraft-to-aircraft” basis, including all types of operations (departures, arrivals and aircraft crossings) that may request runway time. However, despite this level of detail at the runway end, a premise of this research work is that *runway operations plans can be generated and used as a guide for creating gate*

² The ROP problem will be formally defined at the beginning of Chapter 3.

pushback plans (pushback sequence and timing) in a way that enhances airport throughput and delay performance and results in gains in airport operational efficiency, even without managing aircraft taxi operations (from the gates to the runway) at a very detailed level (e.g. assigning intersection priorities) and even without implementing sophisticated surface operations planning schemes (performing advanced taxi route planning). The results and conclusions of this thesis attest to the above statement and constitute the primary research contribution of this work to the field of airport surface operations management.

1.3 Research Approach

With regards to conducting research on Surface Operations Planning in general, the overall approach that was followed, part of which is the work presented in this thesis, consisted of the following steps:

- Research supporting this thesis:
 - Conducting *a thorough study of the structure and the properties of the ROP problem*, in order to identify the numerous relevant parameters involved in it and gain a clear understanding of their highly dynamic behavior, their interaction with each other and their effect on system performance functions, such as departure throughput and delays.
 - Proposing a *conceptual architecture for the Departure Planner system*, which is the decision-aiding tool envisioned to assist air traffic controllers in managing and regulating airport ground traffic (presented in Appendix B).
- Research work presented in this thesis:

- Focusing on the ROP problem for which *a decomposed heuristic algorithm is developed* based on the identified problem properties. Decomposition is chosen because, the complexity of the ROP problem makes it difficult for controllers to consider all of the parameters and details they ought to consider in the context of runway operations planning and therefore, calls for parsing of the problem in separate stages in order to make it simpler to address. The solution generated by the decomposed heuristic algorithm is a runway operations plan, or, in other words, a set of aircraft departure sequences and times, which is based on specific (often competing) objectives and is subject to a number of often very strict constraints.
- Using the runway operations plan generated by the decomposed algorithm as a guide for *planning gate pushback operations based on a methodology introduced for “translating” runway operations plans back to the gates.*
- Performing *an analysis of the behavior of the decomposed algorithm* across specific dimensions, such as airport operational benefits and solution stability and robustness under varying traffic scenarios in order to investigate the impact that the algorithm can have on airport efficiency.

1.4 Overview of the Document

Background on Surface Operations Management and Runway Operations Planning is provided in Chapter 2. Also, in an effort to implement the research approach outlined in section 1.3 for testing the thesis presented in section 1.2, Chapters 3, 4 and 5 present the following three key elements:

- a) In Chapter 3, *an efficient algorithm* for perform runway operations planning is introduced together with the genesis and application of the idea that resulted to this decomposition – based, two-stage Runway Operations Planning (ROP) algorithm, as well as the formulation of each stage of the algorithm.
- b) In Chapter 4, *a methodology for “translating”* the runway operations plans suggested by the ROP algorithm back to the gates is discussed. The objective of this back-propagating “translation” is to generate gate pushback plans which make more intuitive sense to air traffic controllers when they are provided to them as planning suggestions.
- c) In Chapter 5, *a test bed* used to evaluate the performance of the ground operations planning system that is comprised of the ROP algorithm and the gate pushback schedule (sequence and timing) generation procedure is presented, which consists of the benchmark airport model used in the tests and the Matlab/Simulink model developed and used to simulate the two-stage algorithm.

Once the evaluation of the two-stage algorithm behavior is completed, simulation results are presented in Chapter 6. Finally, conclusions from the entire research effort and contributions to the field of airport operations management are discussed in Chapter 7 together with some ideas for the future implications of this work.

An overview of results and lessons learned from the field observations which were performed mainly at Boston’s Logan airport, is given in Appendix A. Also, based primarily on these results, a conceptual architecture for the Departure Planner decision-aiding tool is proposed as a new structured way to view the problem of planning airport ground operations and is presented

in Appendix B. Details about the application of common system design techniques, such as Functional and Relational Decomposition, on the Runway Operations Planning problem can be found in Appendix C. Appendix D presents details on the format of the output of the simulation of the 1st and 2nd stage of the two-stage Runway Operations Planning algorithm, Appendix E shows evidence that the Monte Carlo simulation of the algorithm actually converges and Appendix F shows the results of the standard error analysis that was performed on simulation results to support the statistical significance of some of the conclusions drawn from these results. Finally, a short note on a different research effort for solving runway operations planning problems in one stage can be seen in Appendix G.

Chapter 2

BACKGROUND

Departures and arrivals interact through the common use of airport resources (gates, taxiways and runways). Thus, managing the departure flow at an airport requires an integrated “surface-air” solution that considers all the aircraft on the ground as well as the aircraft in the air that are expected to land during the time period when the departures under consideration are or are expected to be on the airport surface. Therefore, the problem of managing and optimizing runway operations must be posed under the umbrella of the larger problem of managing airport ground operations (Surface Operations Management (SOM) problem).

2.1 Surface Operations Management (SOM)

The tasks involved in Surface Operations Management (SOM) are depicted in Figure 2.1. As shown in the figure, SOM can be divided into the following two main tasks:

- **Surface Operations Plan Generation:** generating feasible and optimal (or near optimal)

plans for utilizing available surface resources among the different types of operations that require service on them (departures, arrival and runway crossings) and

- **Surface Operations Plan Execution:** executing the plans in a safe manner.

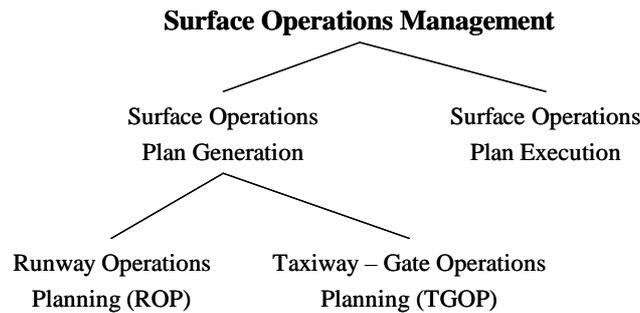


Figure 2.1: Surface Operations Management: task structure

2.1.1 Literature Review: Surface Operations Management (SOM)

Official documentation by the Federal Aviation Administration (FAA), such as the Air Traffic Control Manual [52] and Standard Operating Procedures (e.g. [50]), that contain information on airport ground and terminal area operations and procedures in general, are more descriptive than analytical. They do not provide controllers with algorithms for detailed tasks, such as “slot-to-slot” aircraft sequencing and scheduling that air traffic controllers have to perform while handling ground and terminal area traffic.

While there has been significant research on the modeling and analysis of the airport system in general and the arrival and departure processes in particular, the focus of most of these efforts has been more on the strategic than on the tactical aspects of Air Traffic Management. For

example, *capacity models* such as the ones presented in [48], [55], and [56] attempt to estimate an airport's arrival and departure capacities by estimating the values of various parameters affecting airport capacity. In some cases, capacity models can assist Air Traffic Control operators in the strategic management of air traffic by estimating the effect of strategic arrival and departure capacity allocation decisions [60], [61] and other air traffic management actions, such as Ground Holding [7], [8], [100], [101], [119], [127], [128], [129]. Also, in other cases, strategic capacity models are useful in developing airport infrastructure [9].

There are also *traffic models* to either predict short-term traffic flows [42] or solve ATM problems of a more tactical nature, such as the Aircraft Sequencing Problem (ASP), i.e. finding a good sequence of landing / departing aircraft. The latter problem has been studied in several publications [19], [40], [41], [96], [121], [122], however, in most cases, the departure side of operations was either partially considered or not included at all in the study and the interaction between arrivals and departures was ignored. Horangic [66] proposed queuing models of airport delays, in which typical queues (e.g. $M/E_r/k$) were used to represent only the arrival runway, while Sadoune [105], in his effort to develop the Flight Path Generator and Schubert [107] both tried to include ideas for arrivals and departures in the terminal area. Nevertheless, in the last 10 to 15 years, it is apparent that the focus of research in this area has been primarily on determining flight paths for the arrival flow. This is evident by the fact that decision-aiding tools for air traffic controllers, such as experimental systems developed in Europe [86] or the Center TRACON Automation System (CTAS) developed by NASA [37], [103], [134] in the US, have been primarily developed for assisting controllers with handling arrival flows.

In recent years, research establishments both in the US and Europe, have shifted research attention to the departure flow. Academic research literature documents the wide use of classical *queuing theory* to model airport ground operations and more specifically the departure process. Early predictive models of departure capacity and taxi-out delays were developed by Shumsky in order to forecast takeoff times of departing flights at major airports [109], [110]. Delcaire [43] also used simple queuing representations in order to provide the basis for certain policy suggestions to airport authorities and operators. Hebert and Dietz [62] used five days of data and common probabilistic distributions (Poisson and Erlang) to develop stochastic Markov chain models for predicting departure delays at LaGuardia airport. Pujet [97], [98] extended previous queuing research by developing an input-output “mesoscopic” queuing airport model that integrates the airport terminals, the taxiways and the runway system as queuing servers, in order to provide enough detail to estimate the effectiveness of departure control schemes in reducing taxi-out times and the associated environmental impact. As an extension to Pujet’s work and in an effort to analyze airport ground operations in more detail, the use of queuing theory as a modeling tool for ground operations was further applied to gate operations [6], to the taxi-out process [75] and even to Air Traffic Control flow restrictions in the terminal airspace surrounding an airport [30], [31].

Apart from capacity and queuing models that aim to analyze an airport system and more specifically its ground operations, several *automation systems*, whose core logic is primarily based on optimization methods, have been under development in recent years. For example, the Flight Guidance Institute of the German Aerospace Research Establishment (DLR) developed the Taxi and Ramp Management and Control (TARMAC) system, which performs taxi-route

planning that produces an optimal route from the gate to the runway threshold in order for departing aircraft to meet a specific takeoff time window [21], [22], [45], [125], [126]. This system was based on the concept of “an integrated A-SMGCS” (Advanced Surface Movement Guidance and Control Systems). This concept, according to the pioneers of the A-SMGCS European initiative [46], considers departure planning as an integral part of the overall air traffic management system and dictates that planning must be performed in co-ordination and co-operation with arrival planning, surface movement planning and conformance monitoring [45], in order to successfully support air traffic controllers and pilots in their efforts to manage and optimize the flow of incoming and outgoing traffic on the airport surface.

The Surface Movement Advisor (SMA) started in the 1990s [57], [58], [83] as a joint Federal Aviation Administration (FAA) and NASA Ames Research Center effort. However, it never reached the point of an automated ground operations scheduling function. It is now simply used by the airline ramp controllers at Atlanta airport (ATL) as a tool for information exchange and for observing and monitoring the state of aircraft on the airport surface.

The experiences in all recent tool development efforts indicate that no ground operations aiding tool can be successfully designed without careful field observations and without the direct involvement of all the expected users of such tools at busy airports, such as the airlines, the FAA controllers and the airports operators [38]. Based on this premise, the MITRE corporation developed the Departure Enhanced Planning And Runway/Taxiway-Assignment System (DEPARTS) prototype [36], [44], based on observations of ground operations at Atlanta airport. Furthermore, NASA Ames and Metron Aviation have been developing the more advanced

Surface Management System (SMS) research prototype [10], [11], [12], [25], [112], [113], [117], which is supported by the results of other research initiatives as well ([76], [79]).

2.2 Surface Operations Plan Generation

Planning aircraft operations on the entire airport surface entails the scheduling of aircraft clearances at various control points along the departure flow. One of these control points³ exists at the runway end where aircraft line up awaiting clearance to take off and mix with:

- a) Other types of operations, such as arrivals and crossings that request service on the same active runway and
- b) Other aircraft flying in the airspace surrounding the airport.

Controlling this mixing of departures with all other operations is one of the following two main tasks that Surface Operations Plan Generation is comprised of (Figure 2.1):

- In **Runway Operations Planning (ROP)** the sequence of operations on the runway and the time for each operation in the sequence (i.e. the takeoff sequence and schedule) are determined while uncertainty in pushback and taxi operations is taken into account. Due to different spacing requirements between successive takeoffs dictated by the runway wake vortex separation standards, the takeoff sequence and schedule is a major factor in

³ More on airport surface control points can be found in Appendix A.

determining the efficiency of the runway system⁴. Therefore, in planning runway operations, the final takeoff sequence and schedule are designed as close to optimality as possible in order to achieve specific airport operational objectives and mitigate existing departure inefficiencies related to factors, such as wake vortices, downstream constraints (e.g. traffic flow restrictions, splitting departure routes, jet-prop mix, arrival-departure mix), workload limitations and intersecting runways. At the same time, ROP is a planning strategy⁵ that ensures safe and efficient operations of the runway system, i.e. maintaining wake vortex separations, balancing runway loads, satisfying Air Traffic Control constraints and servicing all operations scheduled to be serviced on the runway system under the current runway configuration as fast as possible. The solution to the ROP problem can be visualized as a Virtual Queue as this was proposed in [1] and described in Appendix B, i.e. a notional waiting line of departing aircraft arranged, at any instant of time, according to the order in which they are expected to take off. In other words, the Virtual Queue can be seen as a virtual extension of the physical takeoff queue that depicts the expected takeoff sequence for all departures under consideration regardless of their current position on the airport surface. The solution of the ROP problem can produce runway schedules (sequences and times) that result in runway queues of reasonable size to minimize emissions and also optimally allocate runway time slots for departures and arrivals and appropriate “gaps” for runway crossings.

- In **Taxi and Gate/Ramp Operations Planning (TGOP)** the appropriate taxi and ramp sequence and schedule are determined in a way that ensures that the takeoff sequence and

⁴ For example, grouping aircraft in order to reduce the occurrence of large separations (e.g. 120 sec of separation between a leading heavy and a trailing small aircraft) in the departure sequence can greatly increase efficiency. Furthermore, runway crossings are also a major inefficiency source and therefore controllers try to increase efficiency by reducing the time crossings have to spend on an active runway.

⁵ ... and “aircraft clearance” is the control tool for implementing the ROP strategy ...

schedule are materialized. TGOP is an important step that can influence the feasibility and optimality of the runway plans generated by the Runway Operations Planning task. Assume that a runway plan is suggested by the Runway Operations Planning function for a group of aircraft, which are expected to call the air traffic control tower requesting to push back from their gates in the near future. If this suggested runway plan is not somehow linked to the gate pushback and ramp movement operations of these aircraft, it is very likely that, due to the uncertainty of aircraft movements on the airport surface, the planned aircraft will reach the runway at a sequence and at times that deviate from what the original runway plan dictated. Therefore, it is not only important that an optimal runway operations plan is generated, but it is also essential that such a runway plan is linked back to the gates, in order to provide the air traffic controllers with planning guidance for the pushback operations.

2.2.1 ROP in the Context of Air Traffic Management (ATM)

A more complete picture of the different components of the Air Traffic Management (ATM) system is presented in the functional interdependency diagram of Figure 2.2 ([2]). The Surface Operations Management (SOM) components of ATM⁶ are light-shaded within the entire ATM boundary (darker shading). As shown in the figure, Runway Operations Planning (ROP) is a subset of the Runway Operations Management (ROM) function, which in turn is a subset of the overall Air Traffic Management System (ATM). The relationships between ROP and other parts of ROM and the ATM system are also shown in Figure 2.2.

⁶ One of these components, the Runway Occupancy Planning process, is the focus of this research.

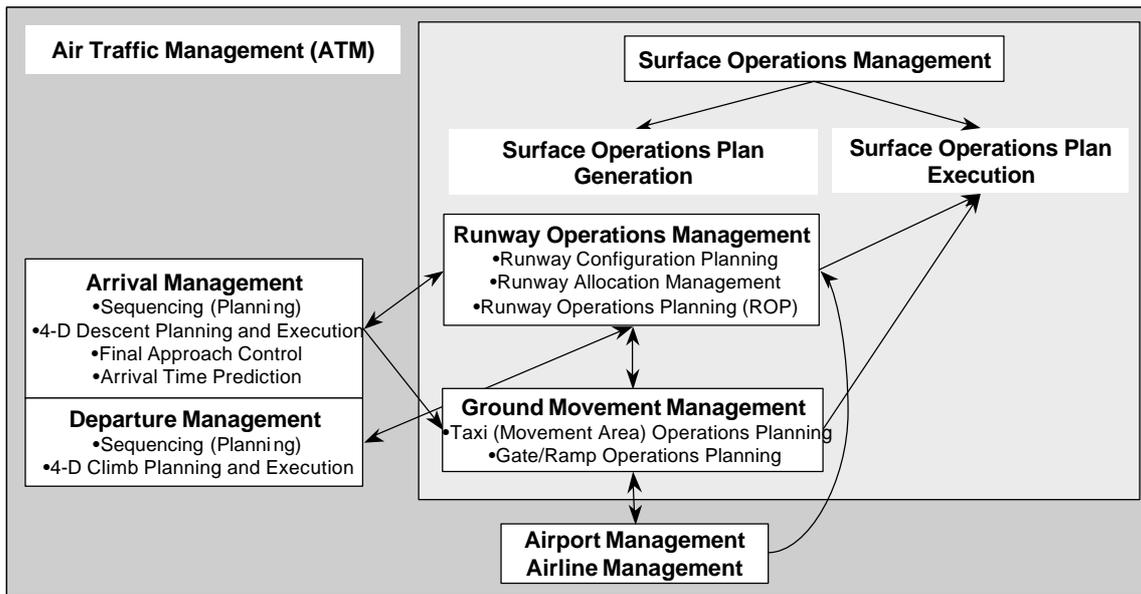


Figure 2.2: Surface Operations Management in the context of Air Traffic Management
(based on ideas by DLR, Germany)

More specifically, the main tasks of ROM are:

- Runway Configuration Management, which involves the strategic planning of:
 - a) Configurations to be used, i.e. the set of active runways and the assignment of different operations on each of them,
 - b) Operational procedures⁷ to be deployed within each configuration,
 - c) Time to initiate a configuration change if this is necessary due to weather conditions, expected traffic fluctuation and other regulations and constraints.

- Runway Allocation Management, which determines the runway to be used by each aircraft.

This assignment is in most cases performed well in advance, while in some cases a re-

⁷ e.g. Accelerated Departure Procedure (ADP) at Boston Logan [73].

assignment may be necessary even if the aircraft has left its gate and is taxiing towards the originally assigned takeoff point and

- Runway Operations Planning, which entails the planning of runway usage for different operations (arrivals and departures) at a more tactical level.

These three tasks are organized hierarchically. Therefore, when thinking about the ROP process, it is assumed that the airport operates under a certain pre-specified airport configuration and each flight has a runway assignment.

Focusing on the Air Traffic Management operations of a single airport, it is obvious that ROM is the connecting link between arriving, departing and ground traffic. Therefore, ROP, as the tactical planning component of ROM, is directly affected by arrival, departure and ground flow constraints. At the same time, ROP produces information that is fed back (or may be fed back in future system configurations) to Ground, Arrival and Departure Management systems.

2.2.2 Literature Review: Runway Operations Planning (ROP)

Several research publications (primarily in Europe) address airport surface operations planning problems through the deployment of *algorithmic methods*, such as greedy search, A* search and genetic algorithms [23], [24], [59]. Teixeira's work at UK's Loughborough University [118] presents a heuristic solution for a "just-in-time" departure scheduler that schedules departing flights, which share a common runway with arrivals. Teixeira actually documented the complexity of the Runway Operations Problem and even though decomposition was one of his

options for reducing complexity, he chose to use separation and linearization techniques for dealing with complexity and solving the problem. He, therefore, separated the global operations problem into an arrival and a departure problem. Assuming that the arrival problem was already solved through established priority principles for arriving aircraft, he attempted to distribute the remaining runway time to departures. Teixeira's heuristic algorithm was an improvement of a greedy (First Come First Serve type - FCFS) heuristic method for locally optimum solutions that scheduled departing aircraft to estimated "gaps" in the arrival traffic given their pre-departure gates and the estimated taxi-out times from gate to runway.

Chang [33] studied algorithmic approaches for planning gate and runway operations. First, he tested heuristic hybrid algorithms and simulated annealing in order to solve the Gate Assignment Problem. Then, he developed heuristic search algorithms (Departure Search, Hierarchical Search and Separate Search Algorithms) to jointly solve the Flight Sequencing and the Gate Assignment problems. Arrival sequencing, gate assignment and departure sequencing are solved in that order, for the arrival and departure aircraft banks ("batches" as called in [33]) at hub airports. As far as departures are concerned, all aircraft ready to depart were sequenced in the descending order of their passenger departure rate, in order to reduce the dwell time (delay) of bigger aircraft, which are likely to have a larger impact on the cost-based component of the objective function that was chosen. However, the drawback of this algorithm is that it may not optimize the runway throughput component of the objective function, because, inter-departure times, which depend on the relative weight classes of two successive aircraft, may not be minimized.

Algorithmic studies of departure scheduling have also been conducted at Imperial College in the

UK and at the Netherlands National Aerospace Laboratory (NLR). Even though early work of the Imperial College research group was mostly geared towards arrivals [15], [16] they recently proposed new research ideas that approach the runway-scheduling problem from both an arrivals and departures point of view [68].

Research studies at NLR resulted in the development of the MANTEA (Management of Surface Traffic in European Airports) Departure Sequencer [63], [64], [65], [120], which determines optimal departure sequences and times and also designs the initial climb phase for each departure. Planning is performed based on a decision support function that is designed to achieve optimal throughput at the runway and at the same time reduce delays and controllers' workload while maintaining safety. MANTEA was developed within the context of the A-SMGCS. In 2000, it was evaluated in two European airports (Paris-Orly and Rome-Fiumicino) and is scheduled for testing in at least three more.

Recently in 1999, the European Organization for the Safety of Air Navigation (Eurocontrol) also started a research initiative for departure operations planning, under the auspices of their European Air Traffic Management Program (EATMP). Several airport visits were conducted [132] and the results on the feasibility of a Departure Manager (DMAN) were reported [87]. In addition, a fairly new Operational Concept of Use for DMAN was developed ([67]).

In the US, departure scheduling research and development of departure decision-aiding tools started with an effort sponsored by the FAA to develop departure-metering software for use in the Southern California TRACON (Terminal Radar Approach Control). This effort led to the

Departure Sequencing Engineering Design Model (DSEDM) tool [99]. This was intended to assist in proving the concept of the Departure Sequencing Program (DSP), by scheduling aircraft to meet Miles or Minutes In Trail restrictions at departure fixes by assigning each aircraft a time window for crossing the departure fix. DSP is now being used at Newark airport (EWR) just as the Airport Surface Management System (ASMS) is operational in Detroit (DTW). Both these tools were fielded in order to assist and improve runway departure operations.

2.3 Research Approach

The research approach followed in the Departure Planner project at the Massachusetts Institute of Technology (MIT) led to the work outlined in this thesis. The Departure Planner project has always been closely tied to Runway Operations Planning since the project's goal is to conceptualize, design and develop a decision-aiding system (the Departure Planner tool) for air traffic controllers, in order to assist them in:

- Optimizing departing traffic,
- Closing unnecessary gaps between arrivals and departures and
- In general, performing their tasks in an integrated Ground Movement Planning System as mentioned in [126].

The Departure Planner decision-aiding tool was envisioned to be a (possibly automated) tactical tool in the sense that the time horizon considered is a short-term horizon of a few hours, with emphasis on the last 30 minutes before take-off.

At busy airports, traffic management of departures includes several control tasks, i.e. pushback, “engine start” time, taxiway entry, runway assignment and takeoff clearances; air traffic controllers exercise these tasks under conditions of high workload and time-critical decision demand. The implementation of a decision-aiding system such as the Departure Planner, can hopefully offer the opportunity to automatically explore a very large number (or even all) possible future departure schedules in order to determine the schedule that best deals with existing uncertainties in the system.

The first step in the development of a decision-aiding tool to improve the departure operations at congested airports is to identify the system constraints that are primarily responsible for generating inefficiencies and delays. This was achieved through an extensive set of field observations of operations at Boston Logan, Washington Dulles and Newark airports (Appendix A). The latter two airports, unlike Logan, are hub airports. Research work thereafter was largely motivated by the lessons learned from these observations. More specifically, once the system constraints were studied, airport system dynamics were analyzed to determine where and how system operations could be adjusted to mitigate the inefficiencies and delays [71], [72], [73], [74]. This process led to a suite of tools that should be included in the Departure Planner system, where in an airport system they should be introduced, and how they should be implemented. The high-level conceptual architecture for this suite of tools ([1]) is described in Appendix B. Each of the system subcomponents presented in this conceptual architecture is designed to exercise planning and control at a particular point along the departure flow. Since the runway was realized to be the most critical “choke point” in the departure flow, the Runway Operations Planning problem was the first to be addressed. A complete problem formulation was developed

including an objective function for the system as well as the appropriate constraints (emanating from Air Traffic Control (ATC) as well as other sources).

The research described in the following chapters addresses the problem of Runway Operations Planning (ROP) by way of a decomposition-based algorithm design methodology, which is implemented to develop a two-stage heuristic algorithm for solving the multi-objective ROP problem.

Chapter 3

DECOMPOSITION-BASED ALGORITHM DESIGN & THE RUNWAY OPERATIONS PLANNING PROBLEM

An “Operations Planning” problem can be formally defined for any type of resource that is used in a production or service process. The resource is associated with a specific time window within which it is available to be used by any type of operation that can be serviced by it. With regards to airport ground operations, field observations showed that three different types of aircraft operations require the airport runway resource: landings, takeoffs and crossings of arrivals taxiing to their gates (stands) and departures taxiing to their assigned runway. Hence, solving the Runway Operations Planning (ROP) problem is basically an optimization procedure that performs the task of “*optimally distributing the available runway resource time among these*

three types of operations” or, in other words, “*generating a sequence and time schedule of which type and which specific operation of that type will occupy the runway at each time point within the available time window*”. The following sections present the methodology that was chosen in this research for addressing the Runway Operations Planning optimization problem, as well as the theoretical background that supports the methodology selected. First, however, a few details on runway usage are presented in order to demonstrate the complexity of the problem of planning runway operations.

3.1 Runway Usage

The runway system is the major flow constraint at an airport. This finding is supported by the field observations at Boston Logan Tower that were presented in [72] and [73]. Therefore, we focus on the cases of runway usage that occur most commonly, excluding special cases such as periodic friction measurement, inspections and emergency cases.

The term “runway occupancy interval”, or, in short, “occupancy”, is defined as the time interval during which a single aircraft occupies a runway. Depending on the airport layout and on the active runway configuration there may be interdependencies between runways so that during a specific time period when one runway is used by an aircraft other runways may not be available to other aircraft. For example, at the German airport Cologne-Bonn, under Configuration 14L, 14R and 25, whenever a landing or a takeoff occupies runway 25, neither runway 14L nor 14R can be used for landings or takeoffs since 25 crosses both of them (Figure 3.1). However, if only

one of the parallel runways is used, only runway 25 becomes unavailable, while the other parallel runway can serve a landing or a takeoff.

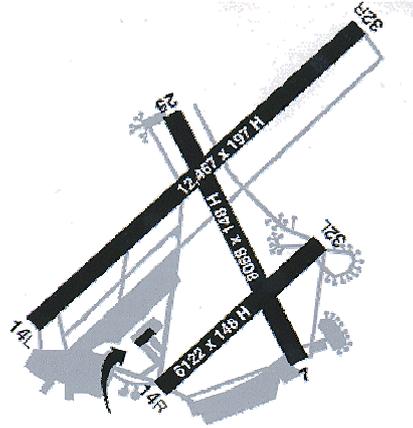


Figure 3.1: Layout of Airport Cologne-Bonn (sketch)

	14L	25	14R
14L	1	1	0
25	1	1	1
14R	0	1	1

Table 3.1: Interdependency matrix for Airport Cologne-Bonn in configuration 14L, 14R and 25.

All such interdependencies can be formally expressed in the form of an interdependency matrix. An example of such a matrix is shown in Table 3.1. Two runway occupancies of any pair of either departing or landing aircraft on the same or on different runways must not overlap when the corresponding element of the interdependency matrix has the Boolean True value (one). On the contrary, an overlap of occupancy intervals for crossings is possible when these crossings occur simultaneously at different points on the runway system. Similarly, an overlap between a

crossing occupancy and a landing or takeoff occupancy is possible when these events occur on different runways, regardless of the corresponding value in the interdependence matrix.

There are two important differences between landing / takeoff and crossing operations. While in all cases there will be a request for exactly one landing / takeoff occupancy interval, depending on the commanded taxi path, departing and arriving flights may or may not have to cross an active runway (and therefore generate request for a crossing occupancy). Furthermore, the nature of crossing operations, in contrast to landings and takeoffs, is such that controllers delay runway-crossing clearances until a group of aircraft has accumulated at various crossing points. At that time, “grouped” aircraft can be cleared to cross one immediately after the other. This option may be used both for planning and control purposes. However, since there usually is limited time between two runway operations, there is usually an upper limit on how many aircraft can be “grouped” at the same crossing point before cleared to cross. In addition, due to limited taxiway space, excessively long aircraft crossing queues can block taxiways and consequently the adjacent runways. In such a situation, landing and / or departure operations may have to be delayed until all crossings are cleared and runways are available again.

In general, it is possible for all three regular runway operations: landings, takeoffs and crossings to be delayed for a certain time, but for safety reasons, arrivals always have priority over departures and crossings.

A complicated set of runway usage rules such as the one described above, coupled with the

uncertainty that inherently exists in airport ground operations rapidly multiplies the complexity of the problem of optimally planning runway operations. As it is a common practice in engineering systems design, complexity calls for a decomposition analysis of the problem in hand, in order to gain insight that will assist in designing a solution algorithm for it.

3.2 Decomposition-Based Design of the ROP Algorithm

An optimization problem, like any engineering design problem, involves a set of functional requirements, a set of design parameters that characterize those requirements and a set of variables that mathematically describe the requirements. Therefore, it is hypothesized that the design of a solution algorithm for an optimization problem can be approached as the design of a new engineering system. Problem decomposition is used in engineering design in order to parse the design problem in simpler subproblems. In a similar fashion, the problem's sets of functional requirements and design parameters can be parsed into two or more subsets and consequently, the algorithm for solving an optimization problem can be separated into independent stages, each of which is easier to formulate in a mathematical program and possibly simpler to solve than the original problem.

In light of the above hypotheses, the ROP problem may be approached from a *more general* “*systems design*” viewpoint as opposed to solving it in a “myopic” way. In that sense, **it is beneficial to view “the process of solving the ROP problem optimally” as similar to “the process of producing an optimal design for a new system”**. For that purpose, techniques that

are common in design theory, such as the Functional and Relational decomposition were applied to the Runway Operations Planning problem. The elements (functions and parameters) involved in ROP were analyzed, with the objective to understand and describe the effect of various parameters on the main problem functions, such as runway throughput and system delays, and the analysis is presented in Appendix C. The end result of this decomposition analysis of the Runway Operations Planning problem is a two-stage solution algorithm. An overview of the algorithm and its primary characteristics is given in the following section.

3.2.1 Overview of the ROP Algorithm

The Runway Operations Planning problem is parsed into two simpler stages as depicted in the flow chart of Figure 3.2. The objective of maximizing throughput is addressed in the first stage together with all factors that affect throughput, such as wake vortex separation and crossing constraints. All other system objectives such as delay minimization and constraints such as downstream flow restrictions (Miles In Trail etc.), workload limitations and intersecting runways are considered in the second stage.

In the first stage of the solution process, one or more feasible sequences of takeoff slots are generated. Each slot is characterized only by the single weight class that an aircraft must have in order to occupy that slot. This is called a “Class Sequence”. The only objective of the first stage is to determine the best (from a throughput perspective) departure weight class sequence (including runway time for crossing operations) to be used in the second stage. This is achieved by calculating the throughput for each class sequence in a family of departure weight class

sequences, which also include “place holders” for aircraft crossing the departure runway⁸.

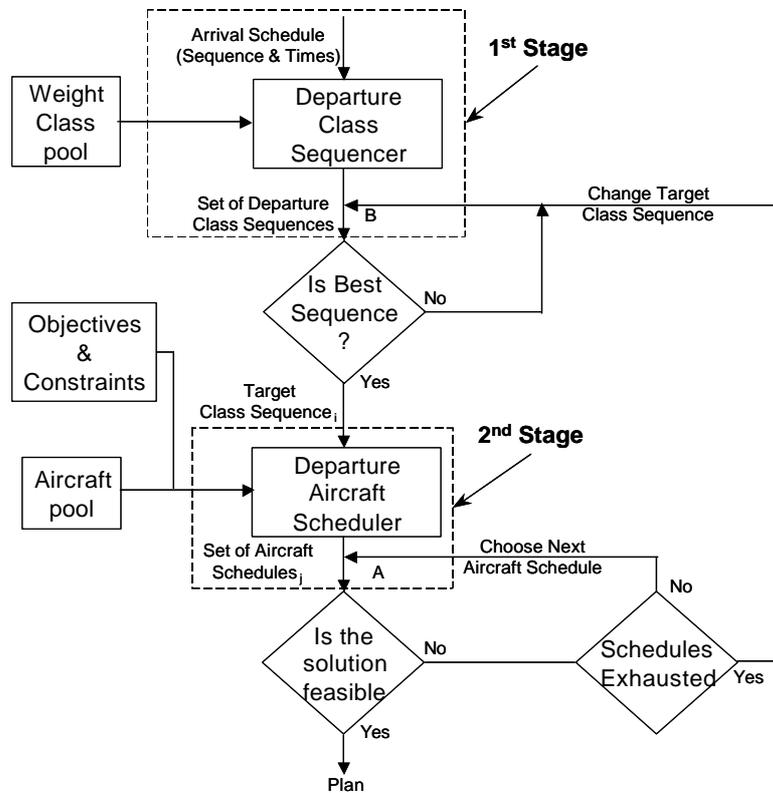


Figure 3.2: Runway Operations Planning in two stages

In the second stage of the solution process, specific aircraft are assigned to the class slots that were developed in the first stage. The resulting series of aircraft is called an “Aircraft Schedule”. The second stage of the algorithm is formulated as an integer program whose solution represents the assignment of specific aircraft to class slots. Throughput maximization is addressed in the first stage of the algorithm and a time-based objective function is used in the second stage subject to all remaining constraints. Together, the two stages of the algorithm perform all

⁸ Scheduling of “place holders” within the departure flow can be also done for arriving aircraft, if dual operations (arrivals and departures) are performed on the runway.

functions required to determine an optimal takeoff schedule (takeoff sequence and times). The class of each departure slot (otherwise referred to as “class slot”) is determined in the 1st stage, and specific aircraft of the correct weight class are assigned to each class slot in the 2nd stage.

3.2.2 Characteristics of the ROP Algorithm

Apart from potential operational performance gains that the featured decomposed algorithm may yield, it also comes with flexibility in two planning stages that can be deployed together or separately. A planning solution can be generated quite fast by performing both stages of the algorithm, the one immediately after the other. However, depending on the needs of the particular real-world situation, the two stages can also be deployed separately. The throughput optimal results of the first stage can be maintained, while the second stage can be solved independently. This way, complete re-solving of the problem can be prevented whenever it is not necessary.

For example, assume that both stages of the algorithm have been performed and a schedule with specific aircraft for each class slot and a specific time for each aircraft has been generated. If one or more of these aircraft have difficulty meeting that schedule, the class slot sequence generated by the first process can be left untouched (especially if it is too costly or impractical to change it), while the second stage (aircraft assignment) can be performed independently to assign new flights to substitute for those aircraft that are unable to meet their class slots. In another example, since throughput is determined in the first stage and the aircraft are assigned to time slots in the second stage, time-based system constraints, such as Expected Departure Clearance

Time (EDCT) slots introduced by ATC, or last minute schedule adjustments to accommodate passenger connections, can be directly incorporated in the optimization without having to consider the potential impact of these time constraints on runway throughput. Therefore, in the presence of sudden system disturbances such as unexpected aircraft mechanical failures or air traffic control restrictions activated at a very short notice, having to change only part of the runway operations plan (as opposed to resolving for a whole new plan) is a way to enhance responsiveness to unforeseen events.

It should also be noted that, the two-stage solution approach for the Runway Operations Planning problem has an advantage in terms of the solution search space over a one-stage solution method⁹. The search space advantage of the two-stage solution methodology comes from the fact that, when the class sequence results of the 1st stage are fixed before entering the 2nd stage, the size of the search space for a solution of the 2nd stage optimization problem is reduced. Assume that there are N departure operations to be planned in total and that N_H of them are of a Heavy weight class, N_L are Large and N_S are Small, where:

$$N_H + N_L + N_S = N \quad (3.1)$$

A one-stage solution method will need to search a space of $N_1 = N! = (N_H + N_L + N_S)!$ potential solutions (assignments of aircraft to takeoff slots). However, if the class sequence is fixed after completion of the 1st stage of a two-stage algorithm, then the N_H Heavy aircraft will only have $N_H!$ possible ways to populate the N_H Heavy class slots in the class sequence and so forth for the

⁹ Even though it is outside the scope of this thesis, for potentially gaining further insight in airport ground operations, it would be useful to compare the results of this thesis to those from a one-stage optimization solution method.

other two weight class categories. Then, the total number N_2 of possible solutions that the 2nd stage of the algorithm will have to search through, is:

$$N_2 = N_H ! * N_L ! * N_S ! \quad (3.2)$$

By looking at the ratio:

$$\frac{N_1}{N_2} = \frac{N!}{N_H ! * N_L ! * N_S !} = \frac{(N_H + N_L + N_S)!}{N_H ! * N_L ! * N_S !} \quad (3.3)$$

we can easily see that for every number involved in the factorial multiplications of the denominator, there is always a number in the numerator that is greater or equal. Therefore, the ratio in Equation (3.3) is always greater than 1. This means that a one-stage solution methodology will always have a larger solution space to search for an optimal solution, than the 2nd stage of a two-stage algorithm.

An air traffic control operator, e.g. the Traffic Management Coordinator (TMC) in an airport tower, can have the opportunity to solve in the first stage a subset of the ROP problem with respect to certain problem parameters that may be more important at the moment, without ignoring the rest of the parameters, but just deferring them to the remaining stage(s) of the process. Also, all controllers (upstream and downstream) may use the class sequence results from the first stage of the planning algorithm as a basis for their tactical decisions. That way, a planning link is provided between upstream and downstream controller positions, however, still allowing flexibility to the downstream controllers (Local Controllers) to exercise control without being as much affected by the decisions of upstream controllers (Gate and Ground Controllers).

In the first stage of the method, one of the most important problem objectives (throughput) is taken into account. Therefore, after that stage is completed, the number of considerations that controllers have to keep track of in their tactical decisions is reduced, with consequent workload reduction benefits. Also, the two-stage algorithm does not in any way prohibit the application of popular controller sequencing strategies, such as jet-prop mixing or alternating exit fixes. These sequencing techniques are mainly driven by downstream restrictions but their application in real-time is in many cases limited by workload considerations. However, it is usually a straightforward task to include such controller strategies in the logic of decision-aiding tools such as the two-stage algorithm presented here.

The structure of the two-stage algorithm described above demonstrates that the parsing of the solution process in two stages was driven by one particular problem element, namely the “weight class” aircraft characteristic. That is because weight class primarily affects the performance function of the ROP problem that was optimized in the first stage, namely the departure throughput performance function.

A detailed mathematical formulation of each stage of the algorithm is presented in more detail later in this chapter, but first, we present the theoretical background based on which, the “weight class” characteristic of each aircraft was chosen as the main driver of the decomposition and the two stages of the ROP algorithm were defined.

3.3 Theoretical Background

In design theory, as developed for the design of electromechanical systems, the functions that a system is supposed to fulfill form the basis for system functional decomposition followed by system synthesis. As stated in the MIT Encyclopedia of Cognitive Sciences [92], “Functional Decomposition is the analysis of the activity of a system as the product of a set of subordinate functions performed by independent subsystems, each with its own characteristic domain of application. It assumes that there are a variety of functionally independent units, with intrinsically determined functions, that are *minimally interactive*. Functional decomposition plays important roles in engineering, physiology, biology, and in artificial intelligence”. It is quite encouraging to be able to identify and it is important to study some of these other areas where decomposition in two or more stages is used as a solution approach to design problems. To this purpose, operations research [18] and design literature ([102], [116]) references were reviewed in order to form an understanding of the current state of the use of decomposition as a design and solution methodology. The main objective of this effort was to investigate the applicability of functional decomposition as a guide for the development of an algorithm to solve the complex optimization problem of Runway Operations Planning. The final product of this investigation, which is an algorithm decomposed in two stages, is based on principles commonly used in the engineering design practice, such as the principles of functional independence and design simplicity.

3.3.1 Literature Review: Decomposition in Design

Most of the descriptions of “design” found in the literature embody the same concept: “design is

a progression from a need to a means of satisfying the need” [102]. Design is understood to be a very essential activity for everyday life and therefore, design methodologies are always thoroughly studied with a purpose of improving their quality. In *engineering design*, decomposition, as a means of achieving an “optimal design” process and/or product, has received early considerable research attention [39] and various decomposition methods have been proposed for two main purposes:

- Solving complex optimization (math. programming) problems that arise during a design and
- Introducing hierarchy and structure in design processes in order to organize product development and reduce design cycle times.

Solving Optimization Problems

Before looking any further, there are several examples that can be drawn from air transportation, such as the “*degradable airline scheduling*” methods currently under research at MIT [80] as well as the techniques used by airlines to solve highly intractable problems such as the *Fleet Assignment (FA)*, *Aircraft Routing and Crew Assignment (CA)* problems. In all cases, multiple objectives and constraints come into play and the only way to reach solutions in a reasonable amount of time is to decompose the problems in separate stages. For example, in the FA airline scheduling stage, an airline maximizes profit by assigning fleet types to flight legs in such a way that they form logically consistent routings (e.g. what flight legs will be covered by a Boeing 737 aircraft). Then, in the Aircraft Routing (or aircraft assignment) stage, specific tail numbers are assigned to specific routes (i.e. a specific Boeing 737 for each route) in order to maximize aircraft utilization. In a similar fashion, the CA first stage generates crew rotations just to satisfy

crew coverage and that solution is used in a second stage where specific crews are assigned to specific rotations in order to maximize utilization without violating all pertinent crew constraints (e.g. crew rest constraints, aircraft-crew compatibilities).

The above examples are related to *large-scale optimization* problems, which are commonly approached in the operations research literature with decomposition techniques. One example of such techniques is the Dantzig-Wolfe decomposition, which is designed for linear programming problems with a special structure and which basically initiated in 1960 the extensive work on large-scale mathematical programming [133]. Another such example is Benders decomposition, which deals with two-stage stochastic optimization problems that involve uncertainty in the second stage of decision-making [18]. Loosely speaking, Benders decomposition is essentially the same as the Dantzig-Wolfe decomposition applied to the dual problem. In order to reduce the problem to a much simpler form, the main approach in both cases involves the decomposition of the original problem into a set of smaller subproblems and a “master” coordinating program. In general, operations researchers have studied extensively the structure of partitioned problems in order to improve computational performance and robustness.

Most of the methods proposed for solving complex mathematical (usually non-linear) programming problems follow steps similar to the Dantzig-Wolfe decomposition. The original problem is decomposed in subproblems, which are heuristically constructed to become independent from each other and are solved independently. Subsequently, the solutions from all subproblems are combined to obtain a solution to the original problem. A comprehensive review of several popular decomposition methods is given in [13]. Most of these methods are

characterized by a two-level hierarchical structure, in which the upper level controls or coordinates the units of the level below [90], [133]. Decomposition is usually achieved by assuming the values of certain global variables fixed in order to eliminate their global effect while cutting design variable links between subproblems [13]. Assigning fixed values to certain parameters is also a method used in model parameter estimation¹⁰ [123]. Through the development of decomposed algorithms, such a problem can be solved in stages. In each stage, a subset of the model parameters is estimated, by minimizing a cost function with the remaining parameters fixed.

Johnson's method [77], [78] is one that does not follow a hierarchical structure but instead is more of an iterative process. The problem is decomposed into a modular component (MC) and an integrating system component (SC) and the vector of problem variables \mathbf{r} is parsed into two components $\mathbf{r} = [\mathbf{u}, \mathbf{v}]$. The subset of variables \mathbf{v} is only involved in the constraint set of the SC, while both \mathbf{u} and \mathbf{v} are involved in the constraint set of the MC. Therefore, the MC is solved first and the solution has the form:

$$u_0 = f_0(v), \text{ where } v \text{ is fixed} \quad (3.4)$$

This solution is then inserted into the objective function of the SC subproblem, which is solved for \mathbf{v}^* . Equation (3.4) is then used in order to find the optimal solution \mathbf{u}^* . In the not so rare case that \mathbf{v}^* cannot be determined explicitly, Johnson's method can be deployed in an iteration process that eventually determines \mathbf{v}^* and \mathbf{u}^* .

¹⁰ Generally a nonlinear programming problem

Wagner and Papalambros [130], [131] studied the decomposition of a given optimal design problem to a set of decomposed subproblems that can be solved independently but coordinated by a master problem. They used an undirected graph representation of the optimal problem to develop a formal methodology that identifies linkages between problem variables or functions. Based on the rigorous identification of these linkages, various structures of the optimization problem at hand can be revealed, which then determine what is the appropriate decomposition (problem partition) to be used.

Several different decomposition methods are used in *mathematics* to simplify the structure of various problems. Specific examples include the Lower-Upper (LU) and the Cholesky decomposition for solving linear systems, as well as the QR decomposition, for orthogonal matrix triangularization [115], since the solution of triangular matrices is easily accomplished by successive substitution in the corresponding linear equations.

Decomposition as an analysis and design tool is also used in *software design*, where, developing software architecture is a problem of managing complexity through decomposition ([5], [108]). Different software design methodologies produce different decomposition results. Also, decomposition is used in *system architecture design* [88], where the following steps are typically performed:

- First, large systems are decomposed to a set of interacting subsystems, each of which provides a related set of services,
- Subsystems may be decomposed further to individual modules and

- Then, in order for this structure to operate as a system, the subsystems have to be controlled so that the results of their operation are delivered to the right place at the right time. Therefore, a framework for subsystem control and communication must be established.

The steps of this process were considered while designing the system conceptual architecture for the Departure Planner decision-aiding system that is presented in Appendix B.

Hierarchy and Structure in the Design Process

In order to introduce hierarchy and structure in design processes, design structure and design incidence matrices are used in the literature to describe the precedence relationships between various design tasks. These tasks are usually grouped in a feed-forward sequence based on the “circuits” of interdependencies between tasks that can be detected. Kusiak and Wang [82] proposed algorithms and Steward [114] proposed matrix transformations (such as triangularization and diagonalization) in an attempt to define input-output relations between design tasks and make the matrices decomposable. Rogers [104] analyzed a design process into design tasks and gathered information about the strength of coupling between the various design tasks. Taking advantage of the coupling strength information, he then developed a rule-based method, which introduces an order in the iterative design process and makes it more efficient, by rendering the design structure matrix triangular.

Ultimately, the objective of the efforts described above is to develop a systematic approach for the decomposition of an overall design process into minimally interdependent tasks, in order to enhance concurrency of the design process and minimize design iterations. Concurrency is

enhanced when the decomposition of the matrices that describe the interdependence and complexity of the problem results in the generation of totally disconnected clusters of design tasks, where no cluster's output is required as input for another cluster. In this case, the design tasks that belong to difference (disjoint) clusters can be performed simultaneously. Nevertheless, the above problem decomposition techniques cannot be applied easily in cases where the problem matrices are hard to construct, because of input-output relations for the different tasks that are difficult to define due to non-existent or unclear causality relations between tasks.

Decomposition is widely used in the field of *supply chain management* and more generally in the field of *manufacturing systems design*. The ultimate goal of *supply chains and manufacturing systems* is to develop products and services that meet specific customer requirements, or in other words “to convert inputs to outputs by processing material” [83]. Given a design target product and a set of customer requirements, several research methodologies have been developed [54] in order to:

- Divide the design target product into smaller, independent subcomponents, each associated with a set of design requirements and
- Decompose overall system requirements into specific design requirements for each system subcomponent and module, in order to perform the design of each subcomponent separately but simultaneously.

The independently designed objects can then be assembled into one entity that meets all customer requirements. Design object (i.e. product) decomposition techniques have been used in order to investigate concurrency in product development [82] and in fact, Prasad [95] suggested

that the level of concurrency can be maximized if a product can be decomposed into components in a way that their dependencies are minimized.

Linck [83], among others ([34], [47], [49], [81]), presents one of the most recent research initiatives in manufacturing and production system design. He used the *Manufacturing System Design Decomposition* technique (MSDD) in order to provide practitioners with a comprehensive manufacturing strategy framework that can guide them through all the stages of the design of a manufacturing system. This approach states the Functional Requirements (system objectives), states the means to satisfy these requirements (Design Parameters) and determines the interrelationships and dependencies among the different elements of a system design. The MSDD approach provides a useful tool for analyzing existing system designs, as well as deriving better alternative designs, as opposed to suggesting local improvements and replacements to the existing ones.

A very important class of problem formulations in the manufacturing system design field is that of *hierarchical formulations* [20], which replace one large production-scheduling problem by a set of smaller ones that are much easier to treat. Hierarchy separation is often based on the fact that different events occur at widely different frequencies, which leads to a *hierarchical decomposition* of scheduling computations. Multiple time-scale problems are studied in the *control theory* [106] for designing system controllers that have different behavior for inputs of varying frequencies. Also, the notion of diagonal dominance for control system transfer functions is used for measuring system coupling in order to produce partitioned transfer functions as they pertain to the design of decentralized control systems [17].

Gershwin [54] suggested the idea of *frequency decomposition*, which determines the frequency of sequential decisions. However, his idea is applied to the design of manufacturing systems. The control of complex manufacturing systems is divided in hierarchical levels, where higher levels are characterized by long time horizons and highly aggregated data, while lower levels have shorter horizons and use more detailed information. Decisions with lower (higher) frequency are grouped into a higher (lower) level decision group (stage). In summary, frequency decomposition groups decisions according to their frequency and imposes an hierarchy to the decision process from a higher level decision group down to lower level decision groups where higher level decisions become inputs for lower level decisions. Simulation experiments are presented in [124] from an application of Gershwin's idea on machine control with a hierarchical controller, which schedules machines at different time scales.

Based on the presence of events and processes that occur at very different frequencies, Caromicoli et al. [27], [28], [29] used multiple time-scale Markov techniques in order to analyze manufacturing systems into a set of models that are simpler than the full model.

3.3.2 The Design Process

As the literature review in section 3.3.1 demonstrates, the design methodologies that have been developed in the past involve a complex iterative and recursive structure. However, most of these methodologies have the drawback that they may not converge quickly to a successful design solution, mostly because they are based on creativity and prior experience as opposed to general principles that enable designers to evaluate design decisions and select good designs

without extensive analysis and experience [102].

Axiomatic design theory claims to address the above design methodology drawbacks. The axiomatic approach to design is an alternative to conventional design improvement techniques and in essence, it attempts to bring an order to engineering design just like the thermodynamic axioms have brought order to the energy field [102]. Developed largely by Nam P. Suh [116], the axiomatic approach offers a set of fundamental design principles in order to enhance the designer's ability to produce good designs and also work in new engineering fields. The two most important axiomatic design principles are:

- a) Designs should be configured to maintain the independence of functional requirements and
- b) Designs should be minimally complex.

The axiomatic design principles map functional requirements and design parameters to each other. In order to produce successful designs, engineers usually look at a design problem through the prism of the following four domains (Figure 3.3) that the design world consists of:

- Customer domain, which is characterized by the customer needs or Customer Attributes {CA} that the customer is looking for in the design of a system,
- Functional domain, which involves the Functional Requirements {FR} that represent customer needs
- Physical domain, which contains the Design Parameters {DP} conceived in order to specify the functional requirements and

- Process domain, which contains the Process Variables {PV} developed in order to characterize the design process.

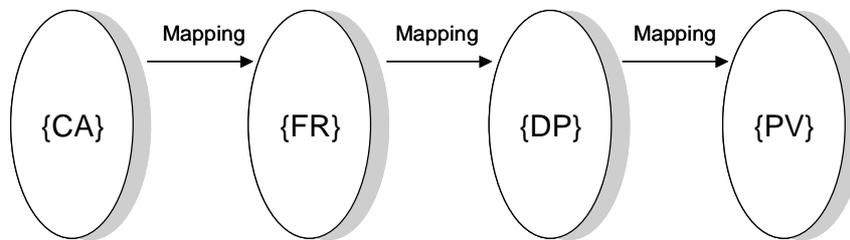


Figure 3.3: Four Domains of the Design World (source [116])

Any design approach involves a “mapping” exchange between the above domains and primarily involves interplay between the functional and the physical domain. The design process is an iterative one because it performs the “left-to-right” mapping shown in Figure 3.3 but also allows the designer to step back to the domain on the left based on ideas and decisions generated in the right domain [116].

The concept of “mapping” between domains brings about the important topic of coupling and decomposition in a design. There is no coupling if a design results in one design parameter affecting only one functional requirement of the design product. “Uncoupled” designs are preferred over other design types because they are characterized by multiple benefits, such as reduced design complexity, easier adjustability and the ability to optimize different parameters sequentially as opposed to simultaneously. In fact, it was shown by Rinderle [102] that optimization of design processes can be accomplished by minimizing coupling in design.

However, all designs are coupled to some extent. Interaction between various facets of a system can be complex and decisions made concerning one aspect of the problem may have ramifications which extend to other aspects of the problem / process / procedure. Considering how difficult it is to create an uncoupled design right from the start, it is very important to determine how much coupling exists in a system. In order to evaluate the degree of coupling, engineers usually perform the following two standard design tasks:

- *Functional decomposition* of a system, which separates the system functional requirements in subsets and maps system parameters and attributes to these requirements and
- *Relational decomposition*, which leads to systems with various functional levels that are independent. Such systems offer a higher chance of robust behavior to failure, because a failure will only affect a certain level of the system and will not propagate to the rest of the independent levels.

The application of the above two design tasks, as well as an example of what each of the domains in Figure 3.3 may contain, is given in the Runway Operations Planning application described in Appendix C.

3.4 Theory: Operations Planning Problem

Imagine a general system for which we have a scheduling problem to solve, where a set of operations have to be optimally scheduled for service at a specific resource of the system. Let:

$$O = \{O_1, O_2, \dots, O_m\} \quad (3.5)$$

be the set of operations to be scheduled for service at the resource and let:

$$A = \{A_1, A_2, \dots, A_n\} \quad (3.6)$$

be the set of attributes that each of these operations has. Each of these attributes A_i can only take a value V_{ij} from a certain set of possible values:

$$V_j = \{V_{1j}, V_{2j}, \dots, V_{kj}\} \quad (3.7)$$

where the number k of possible values is a variable that depends on the type of each attribute A_i .

At an abstract system level, the two main functional requirements of the problem (optimization objectives) are to:

- Maintain or enhance system safety and
- Maximize system performance

Depending on the particular application, each of these two requirements may have more than one way to manifest themselves. For example, maximizing system performance can be:

- Maximizing the number of operations through the resource (throughput maximization) or
- Minimizing the “system traverse” time for each operation from the time an operation enters the system until the point of service completion at the resource (delay minimization) or
- Minimizing the cost of keeping in the system an operation’s request for service before service is initiated and completed at the resource (cost minimization or cost efficiency maximization).

The problem is also governed by a set of system “rules”, which are linked to and have an effect

on system safety and performance through the sequence and timing of operations at the resource. Therefore, the problem to be solved can be posed as: *choose the best sequence and timing of operations at a given resource in order to satisfy all system functional requirements subject to a set of “rules” that are currently active.*

3.4.1 Functional Requirements of the Operations Planning Problem

In most cases, there are functional requirements that can be treated both as an objective and as a constraint and therefore, the designer has to subjectively decide which of these requirements pertain to system performance and which pertain to satisfying system constraints. For example, imagine a problem property that is characterized by the mathematical variable P_1 and is related in a certain way to system performance SP . The higher the value of P_1 is, the higher the value of SP will be and the better the system performance achieved. Obviously a functional requirement of the system design is to maximize system performance. With respect to the particular property, the functional requirement of performance maximization can be expressed either as an objective by maximizing the value of P_1 (probably among others):

$$\dots + \max P_1 + \dots \quad (3.8)$$

or as a constraint by constraining the values of P_1 within a certain acceptable range:

$$P_1 \geq P_{1L} \quad (3.9)$$

where P_{1L} is the lowest allowable value for P_1 .

3.4.2 Design Parameters of the Operations Planning Problem

The set of design parameters includes all those parameters involved in the process of designing a solution for the scheduling problem. These parameters are involved in the characterization of the system functional requirements and are also usually directly linked to specific attributes (members of set A, equation (3.6)) of the system operations (members of set O, equation (3.5)). For example, in the context of the Runway Operations Planning problem that is addressed in more detail in Appendix C, the design parameter “weight class / traffic mix” is involved in the characterization of the system functional requirement of “maximizing runway throughput” and is also directly related to the attribute “aircraft weight class” of each runway operation involved in the problem. Note that, in some cases in the remainder of this thesis, we use the terms “design parameter” and “operation attribute” interchangeably.

For the purposes of this thesis, we deviate from the common use of the term “design parameter”. That is because, we treat design parameters more from a perspective of the entire parameter set which is an integral part of the system design process and we do not consider design parameters to be just those problem parameters whose values are set by the system designer. Some of the problem parameters can indeed be given design values by the system designers, yet some other problem parameters take values as a result of the problem solution methodology. Therefore, we distinguish two types of design parameters:

- *Input Design Parameters*, which offer to the designer / user the freedom to design the solution output and affect system performance by selecting in advance (and modifying as needed) the values of those parameters (e.g. the Maximum Position Shift (MPS) parameter for the Runway Operations Planning problems, see section 3.8.1) and

- *Output Design Parameters*, which are involved in the output solution of the system and therefore affect system performance functions, but cannot take their values in advance like the input parameters do (e.g. the individual delay that each departing aircraft has to suffer as a result of the solution of the Runway Operations Planning problem).

3.4.3 Decomposition of the Operations Planning Problem

As time progresses, operations from the set O enter the system and request service at the resource. Solving the “operations planning” problem requires that operations be scheduled in a way that satisfies the system functional requirements, without violating the set of active system “rules”¹¹ i.e. constraints. This can become a very challenging problem to solve if, in addition, we consider the fact that there are a large number of possible values (some of which may even change in the last minute) that the attributes of each operation can take.

The “quality” of a particular solution (sequence and schedule of operations) in terms of system performance and the degree to which that solution satisfies the problem “rules” (constraints), is directly related to the values of the attributes of each operation involved in the schedule of operations that constitutes the problem solution. Therefore, it seems reasonable to address the “operations planning” problem by using the attributes (set A) of the operations (set O) as the determining factors for the operations sequencing and timing. In the following paragraphs, we will show how, by examining the behavior of problem design parameters that are related to the various operations’ attributes, this approach can be used as the basis to generate an appropriate

¹¹ Some rules may or may not be active at any particular time.

decomposition of the “operations planning” problem in order to reduce problem complexity.

As mentioned earlier, in order to simplify the problem, design engineers traditionally examine the coupling behavior between functional requirements and problem design parameters in order to decompose the problem into smaller subproblems. The results of the functional and relational decomposition analyses assist in describing the relation between the system functional requirements {FR} and design parameters {DP} in the form of a Design Matrix equation, such as the one described in Figure 3.4.

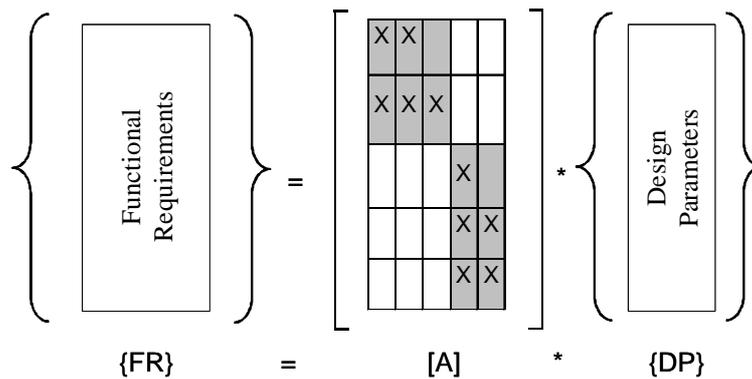


Figure 3.4: Design Matrix equation (example) - Mapping between Functional Requirements {FR} & Design Parameters {DP}

However, it is possible that the functional and relational decomposition described at the end of section 3.3.2 will not yield sufficient “coupling” information based on which, a plausible partition of the problem can be generated. This can happen when there are unclear causality links between some of the problem functions (e.g. system throughput) and some of the problem parameters (e.g. maximum allowable delay for an individual operation). In such a case, some coupling information may be available but it is possible that it will not be enough to determine a

decomposed solution path to be followed for designing an efficient solution algorithm.

The question that arises then is, whether there exist any additional decomposition dimensions, which could possibly complement and enhance the analysis results from the functional and relational decomposition. In order to gain insight for answering this question, we must examine in more detail the controlling mechanism of the problem solution, i.e. the system design parameters and their associated operations' attributes.

Assume that, after analyzing all system design parameters, there is a particular one that satisfies some or all of the following properties:

- a) It is “dominant” in the system, because it impacts one of the system performance functions (e.g. system throughput) more than any of the other system design parameters, i.e. a small perturbation in the value of this parameter can cause a significant change in the performance function value,
- b) It is related to an operation attribute that is more “global” in the system than other attributes, i.e. it is active and takes a value for many or all of the operations involved in the system, as opposed to other attributes that may not exist for all operations and
- c) It changes values at a lower frequency than other system design parameters.

Based on the above assumption, the following design-phase tasks can be performed in addition to the functional and relational decomposition tasks mentioned earlier:

Chapter 3: Decomposition-Based Algorithm Design & the Runway Operations Planning Problem

- “*Degree of impact*” - based decomposition, which determines how “dominant” a certain problem parameter is on the system performance functions and how “global” it is for the set of operations involved in the problem,
- *Temporal (time scale-based) decomposition* of a system, which distinguishes the different time scales under which the various system parameters take values. It is applicable in most dynamic systems, because there is almost always a mixture of low and high frequency characteristics that affect system behavior. In fact, control theory capitalizes on that fact by designing controllers that have different properties and exercise different control logic at different frequencies.
- “*Information certainty*” - based decomposition, which determines how well in advance and at what level the values of various system parameters can be known. In many cases, the level of certainty of the value of each parameter is strongly linked to its frequency element.

The system studies based on the different decomposition dimensions mentioned above may or may not yield the same results. However, when a problem parameter or set of parameters $\{DP_1\}$ is somehow distinguished from the remaining design parameters, then it is possible to parse the set $\{FR\}$ of functional requirements, the set $\{DP\}$ of design parameters and the mapping between them in two disjoint subsets:

$$\{FR\} = \{FR_1, FR_2\} \text{ and } \{DP\} = \{DP_1, DP_2\} \quad (3.10)$$

The members of subset $\{FR_1\}$ are linked to and affected by the particular parameter (or group of parameters) included in $\{DP_1\}$, while $\{FR_2\}$ and $\{DP_2\}$ are the subsets with the remaining functional requirements and design parameters. The member(s) of $\{DP_1\}$ are those identified to be most dominant on certain system performance functions and most global (in scope) and / or

assume values in very different time scales than the rest of the design parameters. These members of $\{DP_1\}$ can therefore be chosen as “the basis of the problem decomposition”. If these members of $\{DP_1\}$ are also uncoupled (i.e. functionally independent) from the rest of the problem design parameters, then the problem can be decomposed into more than one stage. In that case, the system performance function(s) directly affected by the elements of $\{DP_1\}$ can be optimized in a first stage of the solution approach. For the remainder of this document, these elements of $\{DP_1\}$, which are problem parameters with respect to which the first stage optimization is performed, will be called the “**decomposition pivot-element(s)**”.

In a first stage, performance optimization can be carried out with respect to the requirement subset $\{FR_1\}$. The particular design parameter(s) that are directly linked to the functional requirements within $\{FR_1\}$ will take specific values that will remain constant for the remainder of the solution process and therefore, the optimal solution (or set of optimal solutions) generated, will be function(s) of ONLY the remaining design parameters $\{DP_2\}$ whose values have not been set (input design parameters) or determined (output design parameters) yet in this first stage. In a second stage, using the optimal solution(s) of the first stage as a basis, and after setting specific values for all input design parameters, optimization can be performed with respect to the remaining functional requirements in $\{FR_2\}$ in order to determine the values of the remaining output design parameters and complement the results of the first stage optimization to a complete problem solution.

It can be argued that, in a solution methodology with more than one stage, there can be various ways to parse the optimization problem. Depending on the current conditions and assumptions

in the problem, the choice of the decomposition “pivot-element(s)” may vary and that, of course, will affect the form of the decomposition. For example, in the problem of Runway Operations Planning, the presence of air traffic control restrictions may mandate that downstream flow constraints (e.g. Miles In Trail (MIT) or Expected Departure Clearance Time (EDCT) time windows) be used as the “pivot-element(s)”, because they appear to be dominating the planning problem, even though normally they may not be as “impactful” on throughput and delays as the weight class characteristics of each flight in the takeoff schedule¹². System data analysis along the new decomposition dimensions that were proposed can assist a designer in wisely selecting “pivot-element(s)”, by uncovering which of the problem design parameters are more “dominant” on the problem’s performance functions and/or are characterized by low volatility and therefore high reliability to plan based on them, at least in a first stage.

The original problem involved all functional requirements and all problem design parameters. As it was mentioned earlier, there are usually functional requirements that can be treated both as an objective and as a constraint. So, after determining which functional requirements will be treated as objectives, namely $\{FR_{obj}\}$ and which would be included in the problem’s constraint set, namely $\{FR_{const}\}$, the optimization formulation would have the following form:

$$\text{Optimize (max or min) } \{FR_{obj}\} = f(\{DP\}) \quad (3.11)$$

Subject to

$$\{FR_{const}\} = g(\{DP\}) \text{ and } \{C\} = h(\{DP\}) \quad (3.12)$$

¹² A specific example on this is given in section 3.5.1.

where $\{C\}$ is the set of other system constraints that are not related to the system functional requirements. Following the decomposition argument that was introduced above, let:

$$\{FR_1\} = \{FR_{11}, FR_{12}, \dots, FR_{1p}\} \quad (3.13) \text{ and}$$

$$\{DP_1\} = \{DP_{11}, DP_{12}, \dots, DP_{1q}\} \quad (3.14)$$

be the first of the two subsets of functional requirements and design parameters (resulting from the decomposition analysis) correspondingly. The function requirement set $\{FR_1\}$ is parsed into the objective set $\{FR_{1obj}\}$ and the constraint set $\{FR_{1const}\}$, both of which interact only with the corresponding design parameters $\{DP_1\}$. Then, the **first stage optimization problem** takes the form:

$$\text{Optimize (max or min) } \{FR_{1obj}\} = f_1(\{DP_1\}) \quad (3.15),$$

subject to:

$$\{FR_{1const}\} = g_1(\{DP_1\}) \text{ and } \{C_1\} = h_1(\{DP_1\}) \quad (3.16)$$

where $\{C_1\}$ is the subset of the constraint set $\{C\}$ that belongs to the first stage (depending on how the optimization setup of the first stage is posed for each particular problem). Note that, the set of functional requirements for the first stage (set $\{FR_1\}$) may be exhausted in the objective function (i.e. the set $\{FR_{1const}\}$ may end up being empty), but even in this case the remaining problem constraints $\{C_1\}$ are still a function of the set $\{DP_1\}$ (equation (3.16)).

In this first stage, based on the decomposition analysis, there are only a limited number of problem design parameters (set $\{DP_1\}$), which influence the value of the objective function in

equation (3.15). However, that does not mean that the rest of the design parameters have vanished from the problem. Each operation member of the set O of operations to be scheduled (equation (3.5)), always carries all its attributes that affect or are affected by the sequence and timing of that operation in the final problem solution. The only difference now is that, due to the decomposition, the remaining design parameters do not affect the optimal solution of the first stage problem. Therefore, the first stage solution assigns specific values only to the decision variables of the first stage optimization problem and makes the design parameters in the set $\{DP_1\}$ not have any effect on the second stage optimization problem. However, the rest of the design parameters:

$$\{DP\} - \{DP_1\} = \{DP_2\} \quad (3.17)$$

remain to be given specific values (as part of the problem input or as part of the output solution) in the second stage of the solution process.

On one hand, the fact that the decision variables of the first stage have specific fixed values makes the objective function of the second stage simpler. However, on the other hand, the constraint set of the second stage formulation is augmented. That is because the fixed values of some operation attributes from the set A of possible operation attributes (e.g. a subset A_{1j} where $j \in (1, \dots, n)$ and n is the total number of attributes that each operation carries) must be introduced in the formulation as inviolable constraints, since it is obvious that a second stage solution cannot cancel or object the “partial” results of the first stage solution. Therefore, after separating objectives from constraints, the **second stage optimization problem** takes the form:

$$\text{Optimize (max or min) } \{FR_{2obj}\} = f_2(\{DP\} - \{DP_1\}) = f_2(\{DP_2\}) \quad (3.18)$$

Subject to:

$$\{FR_{2const}\} = g_2(\{DP\} - \{DP_1\}) = g_2(\{DP_2\}) \quad (3.19),$$

$$A_{ij} = \{A_1, A_2, \dots, A_j\} = const. \quad (3.20)$$

$$\text{and } \{C\} - \{C_1\} = \{C_2\} = h_2(\{DP_2\}) \quad (3.21)$$

3.5 Application: Runway Operations Planning Problem

The results of the Functional and Relational Decomposition as applied to the Runway Operations Planning problem and presented in Appendix C, reveal the interaction links between the problem's main system functions (functional requirements) and the attributes of the aircraft participating in the problem (design parameters) and provide an initial idea of how the problem can be decomposed to less complex subsets. However, as mentioned on section 3.4.3, additional decomposition dimensions are investigated which can potentially refine the problem parsing into uncoupled subproblems. Sections 3.5.1, 3.5.2 and 3.5.3 discuss these decomposition dimensions and explain how the “weight class” characteristic of each aircraft was chosen as the “decomposition pivot element” for decomposing the Runway Operations Planning problems in two subproblems.

3.5.1 Decomposition Formulation

Focusing only on the strong links that resulted from the relational decomposition arguments in section C.3, the shaded areas in the design matrix in Figure C.4 provide an initial idea for a

potential decomposition of the problem, where the “weight class mix” is used as the decomposition “pivot element”. In this case, the problem formulation at the end of section 3.4.3 becomes:

$$\{FR_1\} = \left\{ \begin{array}{l} \text{Maximize} \\ \text{Throughput} \end{array} \right\} \quad (3.22) \text{ and}$$

$$\{DP_1\} = \left\{ \begin{array}{l} \text{Weight Class /} \\ \text{Traffic Mix} \end{array} \right\} \quad (3.23)$$

The objective function of the **first stage optimization problem** is:

$$\text{Maximize Throughput} = \text{Min } T_{total} = \text{Min } \sum_{m,n \in N_D} T_{m,n} \quad (3.24)$$

where T_{total} is the total time needed to complete the departure sequence as a sum (linear combination) of the time gaps $T_{m,n}$ between successive departure operations (m and n) that belong in the group of N_D available departures. The length of each of these time gaps is a function of the weight class characteristics of the two departure operations right before and right after each gap and therefore, this total time T_{total} is a function of the Weight Class / Traffic Mix design parameter of the problem. Hence, the objective of the optimization becomes:

$$\{FR_1\} = \text{Maximize Throughput} = \text{Min } T_{total} = f_1(\{DP_1\}) \quad (3.25)$$

and it is subject to the following constraints:

$$T_{m,n} \geq SEP_{m,n}, \quad \forall m, n \in N_D \quad (3.26)$$

where $SEP_{m,n}$ is the separation time necessary between successive operations m and n, which is also a function of the Weight Class / Traffic Mix design parameter, based on the entries of Table C.2. Therefore, the constraint set also assumes the form of a function of the set $\{DP_1\}$. Each

optimal solution of this first stage optimization problem is also a function of the Weight Class / Traffic Mix design parameter, because each solution is a sequence of weight class slots that represents a runway operations plan. Each such solution however, is not complete yet, because only the weight class characteristic of each takeoff slot has been determined.

For the **second stage optimization problem**, the set of remaining design parameters is:

$$\{DP\} - \{DP_1\} = \{DP_2\} = \left\{ \begin{array}{l} \text{Aircraft to Slot} \\ \text{assignments} \end{array}, \text{MPS Value}, \begin{array}{l} \text{Individual Aircraft} \\ \text{Taxi Out Delays} \end{array} \right\} \quad (3.27)$$

and the formulation of the optimization problem is:

$$\{FR_{2obj}\} = \text{Min Delays} = f_2(\{DP_2\}) \quad (3.28)$$

subject to:

$$\{FR_{2const}\} = \left\{ \begin{array}{l} \text{Satisfy Downstream} \\ \text{Flow Constraints} \end{array}, \begin{array}{l} \text{Fairness:} \\ \text{Constrain Pos. Shifts} \end{array} \right\} = g_2(\{DP_2\}) \quad (3.29)$$

$$\text{and } A_1 = \text{weight class attribute of each slot} = \text{const.} \quad (3.30)$$

where the constraint in equation (3.30) is dictated by the first stage optimal solution.

The above set of formulations for the two stages were generated based on the selection of the “weight class” as the decomposition “pivot element”, because that is what primarily affects the performance function of the ROP problem that was chosen to be optimized in the first stage, namely the departure throughput performance function. However, there are still research questions to be answered, such as: *is there a unique “pivot element” and if not, then which is the*

appropriate “pivot-element” and under what conditions is it appropriate?

It can be argued that, in a solution methodology that is decomposed in more than one stage, there is more than one design parameter that could be used as decomposition “pivot-element(s)” and therefore, there could be various ways to categorize the information with respect to which optimization is performed in the different stages and the solution results may vary. For example, in the particular problem of Runway Operations Planning, it is possible to use Miles (Minutes) In Trail or other downstream flow restrictions as a pivot element. In this case, only the takeoff positions of aircraft affected by flow restrictions would be planned in an initial stage, with a delay-based optimization objective (for example) and the satisfaction of all active flow restrictions as the only active problem constraint. Then, in subsequent stages, remaining gaps in the takeoff schedule would be filled with the remaining aircraft by solving other optimization problems based on the remaining problem objectives (e.g. throughput maximization) and constraints (e.g. separation criteria).

In order to demonstrate the above concepts, we present here a simple example of Runway Operations Planning and how it could be solved in two different cases where the “pivot element” used is the “weight class mix” in one case and the “Miles In Trail flow restrictions” in the other case. Let us assume that eight (8) departing aircraft (labeled 1, 2, 3, ... , 8) are included in the planning window¹³. Two of them are small (S), one is a heavy (H) and the remaining five are large (L) aircraft. Let us also assume that the expected pushback order of the eight aircraft is as

¹³ For a detailed explanation of the “planning window” and the planning process in general, see section B.2 in Appendix B.

follows:

$$1 (L) - 2 (L) - 3 (L) - 4 (H) - 5 (S) - 6 (L) - 7 (S) - 8 (L)$$

where the symbols in parenthesis are the weight class characteristics of each aircraft.

In the first case, the “weight class mix” is used as the decomposition pivot element and therefore, the first stage of the solution method yields a family of weight class sequences as potential runway operations plans each of which has a runway throughput value associated with it. After ranking those sequences in terms of runway throughput, let us assume that the weight class sequence with the best throughput (top of the list) is:

L	L	L	L	L	S	S	H
---	---	---	---	---	---	---	---

In the second stage of the solution method, each of these weight class slots is populated with a specific aircraft by solving the second stage optimization problem. Let us assume then, that after performing the second stage optimization the final solution has the following form:

2-L	1-L	6-L	8-L	3-L	5-S	7-S	4-H
-----	-----	-----	-----	-----	-----	-----	-----

Case 1

where aircraft 2 and 3 are placed 4 positions apart due to a 20 Miles In Trail active restriction that affects them. This solution will be compared to the solution that is generated for the same problem but in a second case described below, in which a different pivot element is used for the decomposition.

In this case, the downstream flow restriction of “Miles (Minutes) In Trail” is used as a pivot element and therefore, optimization in the first stage of the solution method is centered around satisfying those Miles In Trail (MIT) constraints, without worrying about the weight class

characteristics of each slot in the solution takeoff schedule. If, for example, there is a 20 Miles In Trail restriction (translated to at least 4 takeoff positions apart) that affects only aircraft 2 and 3 and we also assume that flight 2 will take off before flight 3¹⁴, then each row of the following matrix is a possible first stage solution that satisfies the MIT separation between flights 2 and 3:

	Takeoff Sequence Position								
Takeoff Sequence	a	2				3			
	b	2					3		
	c	2						3	
	d	2							3
	e		2				3		
	f		2					3	
	g		2						3
	h			2					3
	i			2					3
	j				2				3

Now we assume that the optimization objective of the first stage is to minimize the cumulative taxi delay that all flights affected by downstream flow restrictions (flights 2 and 3 in this case) will have to absorb. Given the expected pushback sequence mentioned earlier, each of the solutions in this matrix is characterized by a certain amount of taxi out delay that flights 2 and 3 will suffer. Let us assume that, after calculating these delay values and ranking the rows of the above matrix in delay ascending order, we have the following set of possible first stage solutions:

¹⁴ If flight 3 is assumed to take off first, then results will change to the exact symmetrical.

	Takeoff Sequence Position								
Takeoff Sequence	e		2				3		
	a	2				3			
	h			2				3	
	b	2					3		
	f		2					3	
	j				2				3
	i			2					3
	c	2							3
	g		2						3
	d	2							3

where the top row (e) represents a “partial” solution of the Runway Operations Planning problem, which satisfies the given Miles In Trail restriction while at the same time assigning minimum cumulative delay to the flights affected by the restriction.

So, at the end of the first stage we have a family of “partial” solutions in the sense that only two of the eight takeoff slots have been filled with specific aircraft. In the second stage, we examine each of these first stage solutions in order to see if they can yield a feasible optimal solution for the whole problem, by trying to populate each of the remaining takeoff slots with one of the remaining flights (aircraft 1 and aircraft 4 through 8). We can do this in two different ways:

- a) We run a second and a third solution stage that correspond to the first and second stage of the previous case. In the second stage, weight class mix is used as another pivot element and given the first stage solution, throughput is maximized and the third stage focuses on: i) delay minimization for the remaining flights and ii) the remaining problem constraints or
- b) We run ONLY a second stage where the objective is delay minimization among the

remaining flights and we ignore throughput considerations.

For this example, we choose to follow the first methodology (with the three stages in total) and for brevity we focus only on the first two “partial” solutions of the first stage (rows (e) and (a)):

e		2			3		
a	2				3		

For each of the above two solutions, we perform the throughput maximization stage by assigning weight class characteristics in each takeoff slot, while keeping flights 2 and 3 fixed. Let us say that the maximum throughput solution (end of second stage) for each of the above “partial” solutions (which were generated at the end of the first stage) is:

e	L	2-L	S	S	L	3-L	L	H
a	2-L	L	L	L	3-L	S	S	H

Of course, for each first stage “partial” solution there are more than one corresponding second stage solutions but we omit them in order to keep the example short and simple.

In the third and final stage, we assign specific flights to the remaining 6 takeoff slots by solving a delay minimization problem, which satisfies all remaining problem constraints, just as we did in the second stage of the first case of this example. Let us assume that the delay minimizing solution that results from this third stage for each of the two “partial” solutions is:

e	1-L	2-L	5-S	7-S	6-L	3-L	8-L	4-H
a	2-L	1-L	6-L	8-L	3-L	5-S	7-S	4-H

Different than Case 1

Identical to Case 1

It is obvious how the first of these solutions (row (e)) is different compared to the one from Case 1 of the example, while the second one (row (a)) is identical to it. This demonstrates that the

selection of the “pivot element” could affect the final solution of the same Runway Operations Planning problem, because as soon as the chosen pivot element changed from the “weight class mix” to the flow constraint “Miles In Trail”, the problem’s optimal solution also changed.

The main point of decomposing the problem in separate stages is to capitalize on the *assumption* that it is beneficial to use, in a first stage, the information that is currently more “dominant” on the system and hopefully more reliable to use and then, in a later stage, deal with the rest of the information that does not impact the system as much and is also more likely to change. That way, if the latter actually changes, it will not affect the whole result but only certain aspects of it. This will, hopefully, make it easier for airlines to deal with last-minute schedule changes introduced by ground ATC planners. The strength or weakness of any assumptions regarding the decomposition “pivot element” used is tested through the simulation described in Chapter 5 and documented through the simulation results presented in Chapter 6. There can be conditions under which, the most accurate and reliable design parameter may not be the best one to use as a “pivot element” because other problem parameters have (now under the current conditions) begun to play an important and dominant role in the problem. Following that idea, we look further into the additional decomposition dimensions that were introduced in section 3.4.3.

3.5.2 Additional Decomposition Dimensions: Degree of Impact

Among all the design parameters involved in this ROP problem, the “weight class” attribute of each aircraft seems to be the one that globally affects the system of runway operations optimization and is “dominant” on system throughput. That is because a small perturbation with

respect to weight class can have a significant impact on runway throughput performance. Assume for example that a ground operations plan has been generated, which results in a specific level of runway throughput being achieved. If a specific flight in the generated plan is switched with another flight (different aircraft tail number) of the same weight class, then throughput is not affected, but if it is switched with a new flight that is operated by an aircraft of a different weight class than the one it is substituting, then runway throughput can change due to different wake vortex separation requirements between the newly-introduced aircraft and its trailing aircraft. This indicates that the problem parameter “aircraft weight class” is “dominant” on runway throughput and in fact, more “dominant” than the factor “aircraft tail number”.

In addition to the above argument, when comparing the potential of weight class to impact throughput and the potential of each other aircraft operational attribute, we must consider that every aircraft has a weight class associated with it, while not every one of the remaining attributes (e.g. ATC restrictions, fairness constraint) is necessarily active for every aircraft in the system. One such example is the “flow restrictions” operational attribute, which could be a potential “pivot element”, as it was argued earlier. However, such an attribute is not always active for all operations because not every aircraft is impacted at all times by a flow restriction, such as Miles (Minutes) In Trail or EDCTs, as it was also pointed out in Idris’s analysis in [74].

Of course, there are other attributes, such as “aircraft destination” for example, which are always active for each aircraft and can also be considered to have a “global” outreach within the Runway Operations Planning problem. However, all such attributes are linked to the specific flight (aircraft tail number) and as it will be shown in the “time scale” analysis that follows, all

such attributes are characterized by higher volatility and lower reliability in comparison to the “weight class” problem attribute.

3.5.3 Additional Decomposition Dimensions: Time scale and Information Certainty

Based on a hypothesis that, among the various ROP design parameters in the Physical Domain (Table C.3), the information regarding some of those parameters is known with more certainty well in advance than the information associated with the rest of the design parameters, we choose to focus on the particular design parameter of “weight class mix”. We already know from the previous analysis that the latter seems to be the design parameter that is:

- Functionally decoupled from the rest of the problem and
- More “dominant” on the throughput performance function of the ROP problem,

We now hypothesize that departure demand in terms of weight class mix, i.e. the mix of weight class characteristics in the pool of aircraft scheduled to depart, changes value at a much slower time scale and therefore is much more accurate long in advance of the actual departure time than the departure demand in terms of specific aircraft, i.e. specific tail numbers. In other words, a behavior trend such as the one shown in Figure 3.5 is expected¹⁵.

As time elapses from the time a departure plan is generated until the time that the actual takeoffs

¹⁵ This will be tested through data analysis later in this section.

occur the level of prediction accuracy for the values of the departure demand in terms of: i) traffic mix (% of Heavy, Large and Small aircraft requesting departure service) and ii) specific aircraft tail numbers, is expected to increase to values up to 100%. However, the accuracy level for the traffic mix demand should be expected to rise faster than that of the tail number demand and actually settle to a high level value possibly before even the departure plan is “frozen”, i.e. before the time point at which no plan changes can be allowed without seriously disrupting the departure flow.

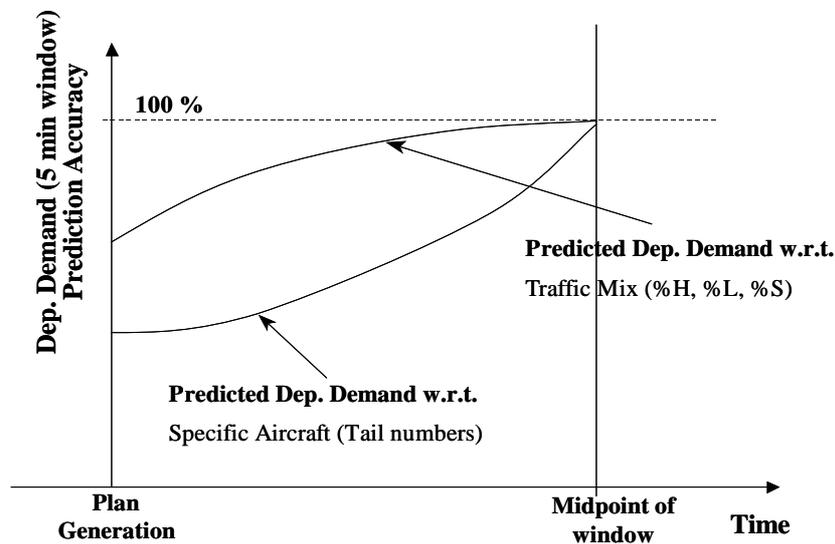


Figure 3.5: Prediction Accuracy of Departure Demand in terms of Traffic Mix vs. Tail Number

Evidence that supports this expected trend is provided through the “Weight Class Mix” and “Tail Number” data analyses that are presented in the following paragraphs. The main data set used for this analysis was the Aggregate Demand List (ADL) data collected from actual operations in the context of the Collaborative Decision-Making (CDM) initiative. Part of the ADL data string

is the predicted pushback time for each departing flight expected to operate¹⁶. This includes flights in the published (OAG – Official Airline Guide) schedule, as well as flights that were not originally scheduled but have nevertheless filed a flight plan. The ADL string is generated (as part of CDM’s real-time operations) every few minutes and is believed to provide the most reliable information about the evolving status of every active or expected air traffic operation. Furthermore, on February 2, 1999, ground traffic data was collected at Boston Logan airport’s Air Traffic Control tower and was used in the analysis that follows. Due to the limitations in space and access to voice communication channels used by air traffic controllers, this set is almost but not entirely complete. It includes the flight information and the time stamps of several key events associated with almost all departure and arrival operations that occurred on that day during the busy morning hours between 6 am and 10 am.

For each of these flights, the OAG-listed scheduled departure time (when applicable) was used as an initial prediction of the pushback time until the first ADL prediction for that flight was generated¹⁷. After that, the ADL data string provided predictions throughout the day at time points referred to as “ADL update times”, which are roughly at every five (5) minutes. These update times were used as the start and endpoints of time windows, which spanned throughout the day. Based on the ADL predictions, each of these windows included a certain number of departures expected to operate within its boundaries. The ADL string also provided the weight class of each aircraft for which a departure time prediction was generated. Therefore, at any

¹⁶ More on pushback time prediction accuracy can be found in [32]

¹⁷ A prediction for a flight expected to operate later in the day, typically does not appear in the ADL string from the beginning of the day, but instead appears only a few hours before the flight’s expected departure time.

update time T_i during the day and for any time window j that spans after time T_i we obtained:

- a) From ADL data, “predictions at time T_i ” % H_{PREDij} , % L_{PREDij} and % S_{PREDij} for the three weight categories of the departure group expected to materialize within time window j and
- b) From ASQP data, “actual” values % H_{ACTj} , % L_{ACTj} and % S_{ACTj} (% Heavy, % Large and % Small aircraft) for the aircraft group that actually pushed back from their gates within the boundaries of time window j .

For any time window j , the “change in % demand” between predictions (performed at time T_i) of the operations expected within time window j and actual operations within time window j can be calculated for each of the three weight categories:

$$\Delta_{H_{ij}} = \% H_{PREDij} - \% H_{ACTj} \quad (3.31)$$

$$\Delta_{L_{ij}} = \% L_{PREDij} - \% L_{ACTj} \quad (3.32)$$

$$\Delta_{S_{ij}} = \% S_{PREDij} - \% S_{ACTj} \quad (3.33)$$

Subsequently, the above three results for all cases that corresponded to the same time difference $\Delta T = T_i - T_j$ between prediction time T_i and midpoint T_j of time window j , were grouped and the mean and standard deviation statistics were calculated for each of these groups. The results for Heavy aircraft demand are presented in Figure 3.6 and for Large in Figure 3.7.

The % change in demand (% of each of the three weight class categories) is shown in these figures as calculated by equations (3.31) – (3.33), from predicted to actual (average values shown with standard deviation error bars), as a function of ΔT . So, for example, derived from all the predictions that were made about 10 minutes before an actual takeoff operation occurred, for

the demand in Large aircraft (Figure 3.7) there was on average an underestimation prediction error of about -2 % with a standard deviation of about 10 %. It seems that, regardless of how far in advance (e.g. 120 min) or close to the actual departure time (e.g. 20 min) a weight class mix prediction is obtained, the average prediction error maintains more or less a small value.

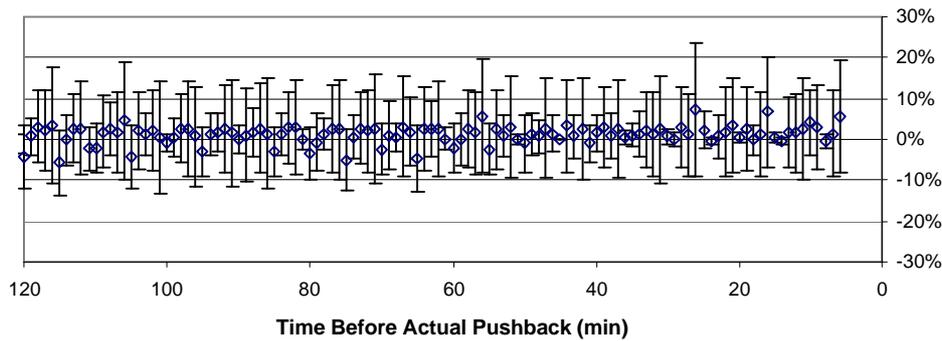


Figure 3.6: % Difference between Predicted and Actual % HEAVIES Demand (BOS, Feb. 2, 1999)

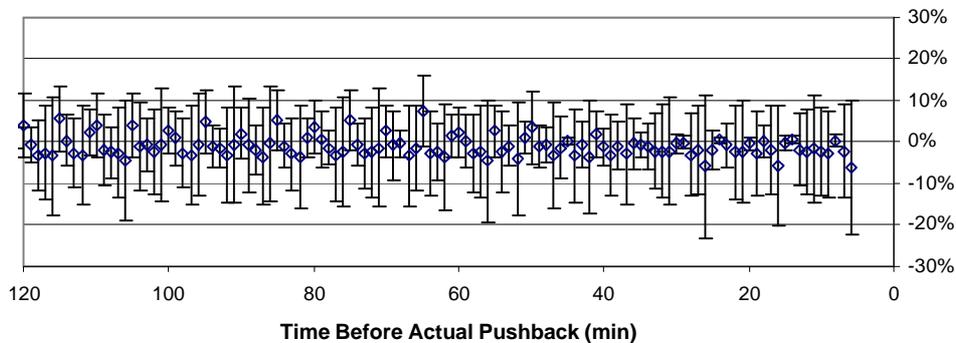


Figure 3.7: % Difference between Predicted and Actual % LARGE Demand (BOS, Feb. 2, 1999)

In order to obtain insight on the behavior of departure demand (pushback time) predictions in terms of aircraft tail numbers, we looked at several flights individually. For each flight in the data set, the actual departure time was collected from Airline Service Quality Performance

(ASQP) data. Results for two flights randomly selected from the February 2, 1999 data set are presented in Figure 3.8 and Figure 3.9. However, many examples such as these could be generated from the same data set.

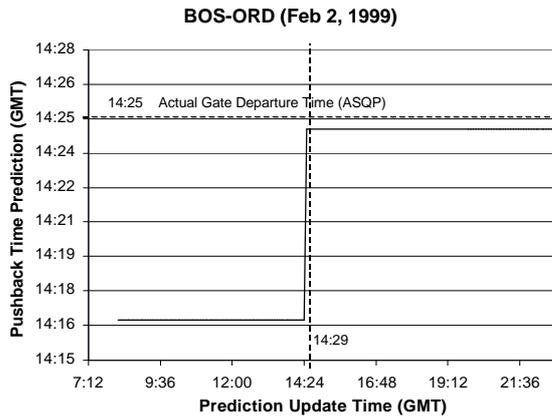


Figure 3.8: Flight Specific Pushback Time Prediction Error: Flight Without Delay

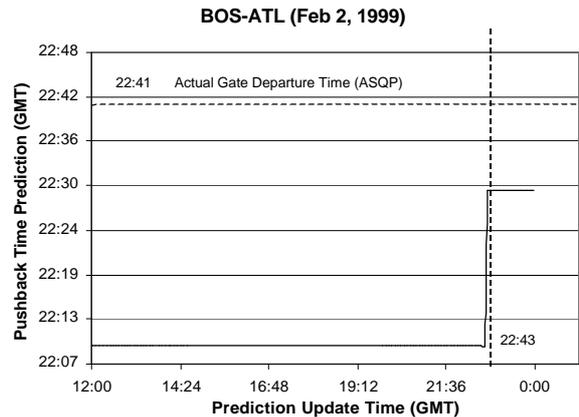


Figure 3.9: Flight Specific Pushback Time Prediction Error: Flight With Delay

Even predictions from one of the best Air Traffic Management tools available, the ADL string, which is believed to provide the most reliable information on every active or expected air traffic operation, demonstrated prediction errors throughout the day when compared to the actual pushback time in both cases and became accurate only after the flight had actually pushed back from its gate. More specifically, the BOS to ATL flight in Figure 3.9 experienced delay, because its actual pushback occurred at 22:41, 31 minutes later than the scheduled pushback time of 22:10. Nevertheless, ADL pushback time predictions were always erroneously equal to the scheduled pushback time (22:10). The aircraft pushed back at 22:41 and only right after that (at 22:43), the ADL predictions increased to a value of 22:30, which was an improved but still not accurate prediction and of course it was provided too late. On the contrary to the previous

example, the BOS to ORD flight in Figure 3.8 did not experience any significant delay. It was scheduled to push back from its gate at 14:24 and it actually pushed back at 14:25. However, ADL pushback time predictions were still not very good in this case either. The ADL data string consistently predicted a pushback time between 14:16 and 14:17. The aircraft pushed back at 14:25 and again after actual pushback (at 14:29), the ADL predictions increased to an accurate value (14:25).

Based on the insight gained from the above analyses, we could argue that departure demand in terms of traffic (weight class) mix, is more reliable in advance of the actual departure time, compared to departure demand in terms of specific aircraft (tail numbers). That is because the “weight class mix” demand changes values at a lower frequency than the “tail number” departure demand and therefore it is more likely to materialize as predicted. Such a behavior is in fact intuitively expected for a number of reasons.

First of all, for any given departure aircraft pool, the attribute “weight class” for each pushback operation has less possible values (only three: H, L and S), while the attribute “tail number” has a number of possible values equal to the number of available aircraft in the departure aircraft pool, which is always greater than three. Therefore, the “weight class mix” problem parameter is likely to demonstrate less variation than the “tail number” problem parameter. Also, especially with the recent increase in regional jets operated at most major airports, the majority of the commercial aircraft that are available to operate flights is of a Large weight class category. Therefore, aircraft schedule position swapping (if deemed necessary) is more likely to involve two aircraft of the same weight class category, even though the tail numbers of the two involved

aircraft will be different. This means that schedule changes, even though they affect the “tail number” demand, they are not as likely to affect the “weight class mix” composition of the departure demand that is expected within a certain time window in the future.

Summarizing, we can say that, complementing all the previous analyses, such as the decomposition and the “degree of impact” analyses, this time-scale study enforced the candidacy of the aircraft “weight class” attribute as a decomposition “pivot-element”.

3.6 Two-Stage Solution Approach

The above analyses looked at:

- a) The “degree of impact” or, in other words, “system dominance” of various problem design parameters and
- b) The time scale under which different departure demand information develops in order to determine how much accuracy there is in each piece of information and when is that level of accuracy reached.

The analyses showed that there is a particular design parameter, namely the weight class mix, which is more robust than other system design parameters and that the corresponding operational attribute of “weight class” has a “dominant” impact on runway throughput and a “global” outreach to all elements involved in the Runway Operations Planning problem. These results

justify the *aircraft weight class* to be the problem “pivot element” on which the first stage of the proposed ROP solution method will be based and the remaining attributes to be treated in a second stage.

The summary of the idea (assuming that a runway is used exclusively for departures and crossings) is: In the first stage, optimization is performed with respect to the more robust set of information, i.e. the weight class mix, and runway operations plans can be generated in the form of sequences of weight class slots without specific aircraft operations assigned to each slot yet. Then, in a second stage, optimization is performed with respect to less robust pieces of information, i.e. specific aircraft demand, in order to complete the results of the first optimization and generate runway plans with specific tail number schedule assignments in each weight class slot from the first stage.

Note that, a similar idea can be expressed in case an active runway is used for arrivals also (mixed operations). In such a case, arrivals can be considered as operations that will occupy certain runway time slots, just like departures do. Therefore, they can also be considered in the generation of class sequences of the first stage. However, new values of required time separations will have to be considered in order to account for the separation standards mandated by the FAA between aircraft pairs that involve arrivals¹⁸. Of course, in this case, the first stage problem to be solved will be more complex. Subsequently, once feasible operation sequences (of weight class slots) are generated, the second stage problem can be solved in order to populate

¹⁸ As opposed to “departure-departure” aircraft pairs only

each class slot in each sequence with a specific departing or arriving aircraft, but in this case, the problem objective function will include the contribution of each arrival to the objective value and the constraint set will include additional constraints generated by the existence of arriving flights in the problem.

The choice of aircraft “weight class” as the particular design parameter of the ROP problem with respect to which the first stage optimization is performed, directly determines (based also on Table C.3) “throughput maximization” as the particular functional requirement that will be used as the objective function of the first stage optimization problem. This can also be seen intuitively. The fact that the traffic mix demand can be known well in advance with more certainty than the specific tail number demand, makes it logical to plan with respect to that specific problem function that is mostly affected by the traffic mix, i.e. runway throughput.

Furthermore, such a separation of dealing with throughput optimization in the first stage and delay optimization in the second stage is encouraged by the results of another type of data analysis found in the literature. Idris [74] performed a thorough analysis of various Air Traffic Control flow restrictions and determined that their impact on the two main system functions, i.e. throughput and delays, can be very different. This analysis required an extensive set of data. However, most of the available databases, such as ASQP and CODAS, which are maintained by the Federal Aviation Administration, were incomplete either with respect to the level of data detail (CODAS) or the type of air traffic operations they record (ASQP). Therefore, a combination of these databases and a set of data collected from real-world operations at Boston Logan airport was used. The two main conclusions of Idris’s analysis were:

- a) Airport THROUGHPUT IS NOT AFFECTED heavily by downstream flow restrictions unless local restrictions (such as Ground Stops) are in effect and in fact many of them (more than two restrictions of the same or different type), however,
- b) Airport DELAYS ARE AFFECTED by both local and downstream flow restrictions. In fact, some restriction types have more significant effect than others. Two different types of delays were examined, namely the “scheduled to actual pushback time” and the “taxi-out” delay, and both analyses yielded the same results [74].

Based on the functional decomposition results that were discussed above and the introduction of the two-stage solution approach, the mapping between the functional requirements and the design parameters of Table C.3 can now be expressed in the form shown in Figure 3.10, if we ignore “weak links”. It is useful to bring the system design equation to that form in order to have an initial idea about the quality of the decomposition. Based on the principles of design theory introduced by Suh [116], the form of the design matrix in a design equation is useful in judging whether the “Independence Axiom” (also introduced in [116]) is satisfied or not, or, in other words, distinguishing a good from a bad design. A good design is considered to be one with a diagonal or a triangular design matrix $[A]$, i.e. “uncoupled” and “decoupled” design in Figure 3.11 respectively [116]. Any other design matrix form results to a coupled design, i.e. (c) in Figure 3.11.

With respect to the shaded and non-shaded areas in Figure 3.10, the design matrix is block diagonal. Hence, the two-stage decomposition that was proposed for the solution algorithm of

the Runway Operations Planning problem seems to be a promising solution approach for this problem.

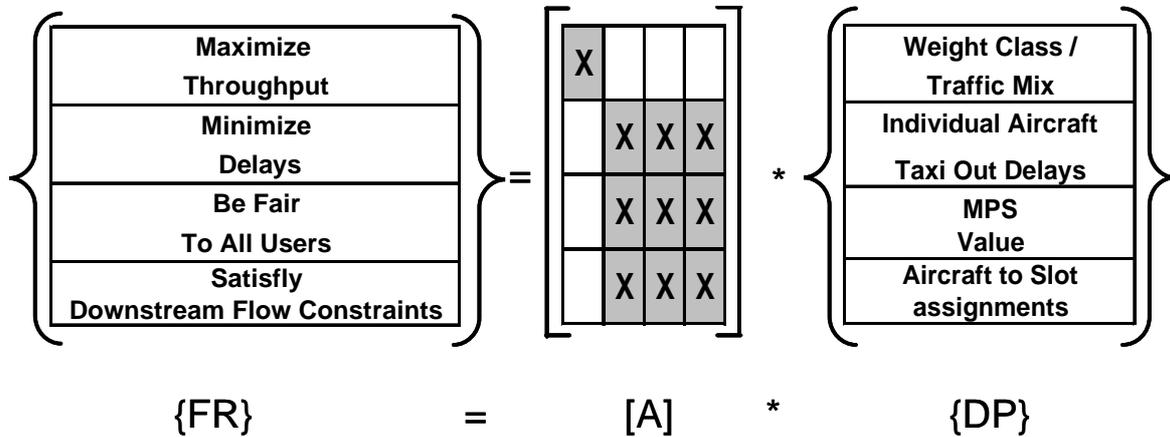


Figure 3.10: ROP – Mapping between Functional Requirements {FR} & Design Parameters {DP}

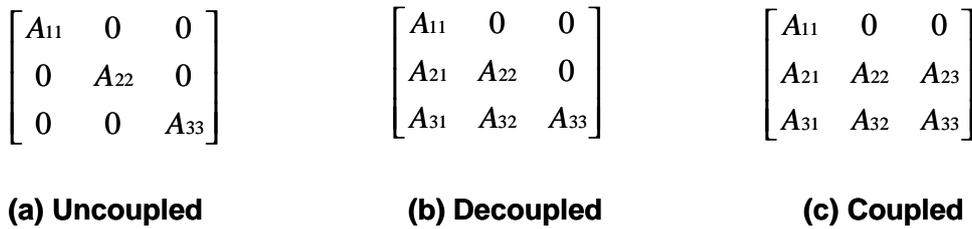


Figure 3.11: Design Quality & Design Matrix

In summary, the structure of the Runway Operations Planning problem was examined through an analysis of the coupling behavior between the problem’s Functional Requirements (FRs) and Design Parameters (DPs). Knowledge of the nature of the particular problem, complemented by specific data analyses, established that, from an engineering design point of view, it could be useful to parse the problem in two subproblems and decompose the solution path in two stages.

As mentioned earlier, Rinderle [102] showed that optimization of design processes can be accomplished by minimizing coupling in design. Hence, the details of the decomposition, i.e. which part of the problem will be addressed in each stage, are determined in a way that minimizes coupling between the two stages. The first stage will handle only the functional requirement of “maximizing runway throughput” while always maintaining safety of operations. The output set with all the solutions from this first stage will be used as the basis for deploying the second stage of the solution. Each possible solution from the first stage will be a function only of certain problem parameters, i.e. a function of the “weight class” characteristics of the departure demand. The remaining “flight-specific” problem parameters, such as aircraft destinations or downstream constraints, will not be part of the solution yet. One of the possible solutions from the first stage will be entered as an input to the second stage and the latter will then be solved with respect to all remaining functional requirements, with the main system objective being the “minimization of cumulative taxi-out delays”. The output of the second stage will take into account all remaining problem constraints and a value will be assigned to the unassigned attributes of each class slot from the first stage solution, such as the tail number, in order to form the complete solution to the overall problem.

The formulation of the two-stage solution methodology is presented in more detail in the following sections. In this formulation, the problem is decoupled and departures are isolated from arrivals. However, the generality of the formulation is not lost if there are runways on which “mixed” operations (arrivals and departures) are allowed. In the scope of this research, it is simply assumed that arrivals are not to be scheduled and that the arrival schedule cannot be altered. Arrivals can therefore be added in the formulation if they are treated as fixed requests

for runway time on runways used for mixed operations. In the following sections, we:

- a) Outline the formulation (objectives and constraints) of the optimization problem addressed in each of the two solution stages and
- b) Describe the core element of each stage and the functions associated with them.

3.7 Stage One of the Solution Methodology

The goal in the first stage is to maximize departure throughput. This is formulated as a linear program and the decision variables are the weight class WC_i associated with each departure slot in the sequence (one decision variable per slot). Each decision variable can take three possible values: one (1) for Heavy (H), two (2) for Large (L) and three (3) for Small (S).

3.7.1 Constraints

The first constraint that affects the separation between successive aircraft is the minimum separation requirement imposed by air traffic control on successive runway operations (departures and / or arrivals) due to wake vortex considerations. These separation requirements are the set of separation times or distances between successive operations on the same runway, whether these are successive departures, successive arrivals, a departure followed by an arrival, or an arrival followed by a departure. As far as successive departures are concerned, for all “leading-trailing” pairs of departing aircraft, the set of separation requirements is a parameter that can be input in the planning system as a square matrix, with row coordinates corresponding

to all possible weight classes for the leading aircraft and column coordinates corresponding to all weight classes for the trailing aircraft (the values used in this document are shown in the matrix in Table C.2). If necessary, each entry of the matrix may be modified as desired by the planner, but in general the values used from Table C.2 correspond to values dictated by FAA regulations and typically used by air traffic controllers.

Assuming that arrivals on another runway must cross an active departure runway in order to taxi to their gates, two additional constraints that can affect the separation between successive departures and therefore the departure runway throughput are:

- The limit on successive arrivals that may be accommodated on an arrival runway if the arrivals on that runway must cross the active departure runway, i.e. the number of arrivals accommodated between successive runway crossing clearances must be less than or equal to the holding capacity of the taxiways segments between the arrival and departure runways and
- The maximum delay that an aircraft waiting to cross can absorb.

These latter two constraints influence departure runway throughput by affecting the times when departures will have to be interrupted in order for crossings to occupy the active departure runway. The limit values for both of these constraints (max number of aircraft or max number of delay minutes) can be input to the planning system as system parameters the value of which can easily be adjusted by the planner.

3.7.2 Objective Function

An easy way to formulate the objective of maximizing throughput is by minimizing the time when the last operation is completed, i.e. has cleared the runway. The formulation of the objective function is as follows: Let N_A be the total number of arrivals and N_D the total number of departures considered. Then, $N_A + N_D = N$, is the total number of “mixed” operations on the runway(s) during the current scheduling window. Therefore, maximizing departure throughput can be achieved by minimizing the time of the latest takeoff $\max t_{D_i}$ ¹⁹:

$$\text{Max departure throughput: } \min \max t_{D_i}, \text{ where } 1 \leq i \leq N_D, \quad (3.34)$$

and maximizing total throughput (of all operations) can be achieved by minimizing the time of the latest “runway operation” (departure, arrival or crossing) $\max t_i$:

$$\text{Max aggregate throughput: } \min \max t_i, \text{ where } 1 \leq i \leq N_A + N_D \quad (3.35)$$

Another way to formulate the objective is by mathematically expressing the total time T_{total} needed to complete the departure sequence as a sum (linear combination) of the time gaps $T_{m,n}$ between successive departure operations (m and n):

$$T_{total} = \sum_{m,n \in N_D} T_{m,n} \quad (3.36)$$

Maximization of Runway Throughput (RT) is then achieved by minimizing this total completion time in order to minimize the overall “separating impact” of wake vortex separation standards on runway throughput:

¹⁹ For the purposes of this formulation, time units are in seconds from the schedule start time, which is equal to zero (0).

$$\max RT \Leftrightarrow \min T_{total} \Leftrightarrow \min \sum_{m,n \in N_D} T_{m,n} \quad (3.37)$$

The length of each of the time gaps involved in the objective function depends on the value of the decision variable (weight class) WC_i associated with each departure slot i as well as that of the preceding slot $i-1$. The exact form of the objective function is based on the values of the wake vortex separation criteria (in seconds) derived from Table C.2, for each “leading-trailing” pair of departing aircraft:

$$T_{m,n} = f(WC_i, WC_{i-1}) \quad \forall \text{departure } i \in [2, \dots, N_D] \quad (3.38)$$

3.7.3 The Departure Class Sequencer

The core element of the first stage is the Departure Class Sequencer. It generates a family of departure Class Sequences, where each of the slots is characterized by an aircraft weight class value (Heavy, Large or Small) that determines the only class of aircraft that is permitted to populate that slot. The function that the Departure Class Sequencer performs is to design the Class Sequences in a way that maximizes departure throughput.

One of the basic assumptions in this module of the algorithm is that the ability to predict accurately enough the sequence and touchdown times of the arrival stream has improved significantly in the last few years [37]. Depending on the runway geometry and inter-dependence, some or all of the expected arrivals, after landing and deceleration, can become runway-crossing requests on an active runway. The times of these crossing requests can be

estimated based on the weight classes of the arriving aircraft (different aircraft have different runway occupancy requirements) and the taxiway space constraints at the specific airport. In many instances, the runway that these arrivals must cross is a departure runway. Thus, maximizing throughput requires the appropriate sequencing of departures and runway crossings.

3.7.4 Output

The output of the first stage is an $m \times n$ matrix CS of Class Sequences (equation (3.39)), where m is the number of sequences generated and n is the size of each sequence, i.e. the number of runway slots that are being planned. These sequences are listed in order of throughput, so each row is a class sequence and the row number reflects the ranking of that specific class sequence relative to the other sequences. The best sequence is listed in the first row. Therefore, the matrix has the following form:

$$\text{CS} = \begin{bmatrix} \text{Class Sequence 1,} \\ \dots\dots\dots \\ \text{Class Sequence } i, \\ \dots\dots\dots \\ \text{Class Sequence } m \end{bmatrix} \quad (3.39)$$

3.8 Stage Two of the Solution Methodology

The first (best in throughput) of the class sequences from matrix CS (output of the first stage) becomes the Target Class Sequence (TCS). The goal of the second stage is then to assign specific aircraft to the weight class slots in the TCS while meeting all other active constraints. If the selected TCS cannot yield feasible solutions, the next best member of CS is set as the new

Target Class Sequence and so forth.

The second stage optimization is performed by formulating the problem as an integer program (IP) that assigns specific aircraft to each class slot and then solving the IP with the Branch & Bound method. As it will be shown in section 3.8.2, the problem can be linear or non-linear, depending on the chosen values of some problem parameters. The decision variables selected for the formulation are X_{ij} , where $X_{ij} = 1$ if aircraft i occupies slot j , and $X_{ij} = 0$ otherwise.

3.8.1 Constraints

One of the most fundamental constraints in the assignment of aircraft to slots is the requirement that no aircraft is assigned to a slot that it cannot physically fill i.e. the slot is earlier in time than the earliest time the aircraft can reach the runway. For example, if the earliest time that aircraft i is expected to be at the runway is time 900 and the time at the midpoint of the first two slots in the TCS is earlier than time 900, aircraft i cannot be allowed to occupy slots 1 and 2 in the final solution. This type of constraint can be easily formulated as $X_{ij} = 0$, for $j = 1, 2$.

The *class slot sequence* of the Target Class Sequence must also be satisfied. So, for example, in a TCS with 10 slots, if aircraft i is a large, it can only occupy large class slots in the sequence.

This can be guaranteed by introducing the constraint

$$\sum_{j \in L} X_{ij} = 1, \forall \text{ Large aircraft } i \quad (3.40),$$

where L is the set of large class slots in the Target Class Sequence. Therefore, for example, if slots 2, 3, 4, 5 and 6 are the “large” slots in the sequence and aircraft 1 is one of the Large aircraft in the aircraft pool, the following must be true in the final solution: $X_{1j} = 0, \forall slot j \in [1,7,8,9,10]$. This can be guaranteed by setting the constraint $\sum_j X_{1j} = 1$, where $j \in [2, 3, 4, 5, 6]$.

Furthermore, given that the number N_S of slots to be populated in the class slot sequence is equal to the number of departing aircraft N_D in the group that is being planned, each slot must be occupied by exactly one aircraft:

$$\sum_{i=1}^{N_D} X_{ij} = 1, \forall slot j \quad (3.41)$$

The obvious constraint that each aircraft must occupy only one slot can be expressed as:

$$\sum_{j=1}^{N_S} X_{ij} = 1, \forall aircraft i \quad (3.42),$$

but in this case such a constraint is redundant, because it is covered by the constraints in equations (3.40) and (3.41).

Various types of *ATC operational constraints* may restrict the sequence position and time that an aircraft can be released for takeoff. In typical airport operations, air traffic is often constrained by flow restrictions. Miles In Trail spacing and DSP (Departure Spacing Program) or EDCT (Expected Departure Clearance Time) time windows are common types of flow restrictions issued by various Air Traffic Control (ATC) authorities such as the FAA System Command

Center or an en-route Air Traffic Control Center that controls several adjacent en-route airspace sectors. When active, air traffic flow restrictions affect only specific aircraft depending on their destination, departure fix or other special flight circumstances, unlike other restrictions, like position shifting constraints introduced for ensuring fairness among users, which typically affect all departing aircraft that are involved in the planning cycle. However, active flow restrictions can introduce significant throughput and delay inefficiencies to an airport system even though they may not affect all departure operations at that airport [74].

Among the most frequently used ATC operational constraints are Miles In Trail (MIT) and (less frequently) Minutes In Trail (MinIT) constraints that impose aircraft separations en route. They can be re-stated in terms of time separation at the takeoff point:

$$\left| t_{D_i} - t_{D_k} \right| \geq \Delta T_{ik} \quad (3.43)$$

where ΔT_{ik} is the minimum time separation at the takeoff point between flights i and k , which have an In-Trail restriction imposed on them. This means that aircraft i and k can only take off at least ΔT_{ik} time units apart in order to ensure that the In-Trail separation is not violated when they will be airborne. More conveniently for this model, MIT constraints can be stated in terms of a minimum required takeoff sequence position separation ΔP_{ik} between flights i and k , which have an In-Trail restriction, imposed on them:

$$\left| \sum_{j=1}^{N_S} j^* X_{ij} - \sum_{j=1}^{N_S} j^* X_{kj} \right| \geq \Delta P_{ik} \Leftrightarrow \left| \sum_{j=1}^{N_S} j^* (X_{ij} - X_{kj}) \right| \geq \Delta P_{ik} \Leftrightarrow \begin{cases} \sum_{j=1}^{N_S} j^* (X_{ij} - X_{kj}) \geq \Delta P_{ik} \\ \sum_{j=1}^{N_S} j^* (-X_{ij} + X_{kj}) \geq \Delta P_{ik} \end{cases} \quad (3.44)$$

This means that aircraft i and k must take off at least ΔP_{ik} takeoff slots apart from each other to

ensure that the In-Trail separation is not violated when they become airborne.

In the case of Expected Departure Clearance Time (EDCT) or Departure Sequencing Program (DSP) constraints, the constraint time window is:

$$tEDCT_{i_1} \leq t_{Di} \leq tEDCT_{i_2} \quad (3.45)$$

or

$$tDSP_{i_1} \leq t_{Di} \leq tDSP_{i_2} \quad (3.46),$$

where t_{Di} is the takeoff time of departing aircraft i , and $tEDCT_{i_1}$, $tEDCT_{i_2}$, $tDSP_{i_1}$ and $tDSP_{i_2}$ are the time values that determine the EDCT time window (typically a 15-minute window [74]) or the DSP time window (typically a 3-minute window [74]) as defined by ATC for flight i . Assuming that the expert input of air traffic controllers is available, a heuristic methodology can

be inferred to translate a takeoff time window, e.g. $[tDSP_{i_1}, tDSP_{i_2}]$ to a takeoff slot window, e.g. $[sDSP_{i_1}, sDSP_{i_2}]$. The takeoff position s_{Di} of any aircraft i ($i \in N_D$) can be written as a

function of the decision variables $\sum_{j=1}^{N_S} j^* X_{ij}$ (N_S is the number of slots to be populated) and the

above constraints can then be formulated for any given aircraft i in the form of an acceptable slot range, as follows:

$$sEDCT_{i_1} \leq \sum_{j=1}^{N_S} j^* X_{ij} \leq sEDCT_{i_2} \quad (3.47) \text{ or}$$

$$sDSP_{i_1} \leq \sum_{j=1}^{N_S} j^* X_{ij} \leq sDSP_{i_2} \quad (3.48)$$

where $sEDCT_{i_1}$, $sEDCT_{i_2}$, $sDSP_{i_1}$ and $sDSP_{i_2}$ are the takeoff slot boundary (upper and lower) index values that define the EDCT or DSP takeoff slot window. Based on their experience in applying air traffic control restrictions when they create mental models of how they want the takeoff sequence to be, air traffic controllers decide for any given flight i what these upper and lower boundary slot values must be.

Lifeguard flights or other types of *priority constraints* can be similarly modeled in the form of an upper bound $P_{\max TO_i}$ on the takeoff sequence position:

$$\sum_{j=1}^{N_S} j * X_{ij} \leq P_{\max TO_i} \quad (3.49)$$

or in terms of inequality constraints between two different flights i and k :

$$\sum_{j=1}^{N_S} j * X_{ij} \leq \sum_{j=1}^{N_S} j * X_{kj} \Leftrightarrow \sum_{j=1}^{N_S} j * (X_{ij} - X_{kj}) \leq 0 \quad (3.50).$$

At many airports, localized *sequencing constraints* also affect the departure efficiency. For example, back-to-back departures to the same departure fix are generally not allowed because they require additional gaps between flights. Typically these gaps are achieved by alternating jet and propeller aircraft departures on the same runway because these two different types of aircraft typically use different departure fixes after takeoff. Such constraints can also be introduced in the form of a position constraint (acceptable departure slot positions for each flight). In the simulation described in Chapter 5, such transitions from time constraints to takeoff slot number constraints were based on our experience in collecting operational data at Boston's Logan

airport, as well as on personal interviews with experienced air traffic controllers.

In many cases, maintaining departure fairness among airport users is a difficult task for air traffic controllers. One possible way to achieve fairness is to introduce a “fairness” constraint through the use of a “*Maximum (takeoff) Position Shifting*” (MPS) constraint that limits the deviation from a “First Come (Call Ready for Pushback) First Serve (Release to Take Off)” policy, unless different specific agreements (known to the optimization planning tool) exist between ATC and the airlines. The MPS value can be predetermined by ATC and the airlines. Based on scheduled or “expected to call ready” pushback data, an expected pushback sequence is formed and each aircraft has its own pushback sequence number. The MPS value then determines the range of acceptable takeoff sequence positions for each departure. If P_{PB_i} is the pushback sequence position of aircraft i and P_{TO_i} is its takeoff sequence position, the MPS value is used in the following constraint:

$$|P_{PB_i} - P_{TO_i}| \leq \text{MPS} \Leftrightarrow \begin{cases} P_{PB_i} - P_{TO_i} \leq \text{MPS} \\ -P_{PB_i} + P_{TO_i} \leq \text{MPS} \end{cases} \Leftrightarrow \begin{cases} -P_{TO_i} \leq \text{MPS} - P_{PB_i} \\ P_{TO_i} \leq \text{MPS} + P_{PB_i} \end{cases} \quad \forall \text{ aircraft } i \quad (3.51)$$

The takeoff position P_{TO_i} can be written as a function of the decision variables X_{ij} as

$$P_{TO_i} = \sum_{j=1}^{N_s} j * X_{ij} \quad \forall \text{ aircraft } i \quad (3.52)$$

and therefore the above constraints become:

$$|P_{PB_i} - P_{TO_i}| \leq \text{MPS} \Leftrightarrow \begin{cases} -\sum_{j=1}^{N_s} j * X_{ij} \leq \text{MPS} - P_{PB_i} \\ \sum_{j=1}^{N_s} j * X_{ij} \leq \text{MPS} + P_{PB_i} \end{cases} \quad \forall \text{ aircraft } i \quad (3.53)$$

where MPS and P_{PB_i} are constants that are known in advance.

3.8.2 Objective Function

Maximizing throughput was the main objective in the first stage of the algorithm. The second important objective in terms of enhancing airport system efficiency is the minimization of delays²⁰, subject to all of the above constraints that apply in each particular planning situation. So, a delay-based objective function is used in the second stage of the algorithm in conjunction with the remaining system constraints. We assume that the time assigned to each runway event is set equal to the midpoint of the time slot that the specified aircraft is assigned to and that the Target Class Sequence with crossings is:

$$L - L - H - s/s/h - L - S - S - H - I - L - L - S$$

where uppercase and lowercase letters signify departure and crossing operations respectively and the letters H, L and S denote the three different aircraft weight class categories (Heavy, Large and Small) considered in this problem. If, for example, the absolute earliest time that a large aircraft from the departure pool can be at the runway is estimated to be 670 (in seconds from time zero, which is at the start of the scheduling effort), then the following set contains the times that correspond to the midpoints of the slots in the Target Class Sequence:

$$700 - 760 - 820 - (\mathbf{X}) - 930 - 990 - 1050 - 1110 - (\mathbf{X}) - 1230 - 1290 - 1350$$

where (X) is used to briefly denote a group of crossing operations (three in the first group and one in the second group) that occurs at that time point in the schedule.

²⁰ The exact definition of “delay” for each operation will be given later in this section.

These calculations are based on wake vortex separations and landing / crossing runway occupancies. For example, if the first Large aircraft (L) is available at the runway to take off at time 670, then, for safety reasons, a gap of 60 seconds is necessary before the next aircraft in the schedule (a Large) is cleared to take off (Table C.2). This gap takes us to time 730 and therefore the midpoint of that time slot is set to $(670 + 730) / 2 = 700$. Similarly, it is calculated that the next aircraft (a Heavy) can be cleared to take off at time 790 (provided that it can be available at the runway end) and the corresponding slot midpoint for the second aircraft is $(730 + 790) / 2 = 760$. Assuming an additional 60 sec that the Heavy aircraft occupies the runway during takeoff, takes us to time 850, which is the first time that the awaiting crossings can be cleared to cross the departure runway. The midpoint of the Heavy departure slot is now $(790 + 850) / 2 = 820$. We assume that there are two crossing points available (X1 and X2) and that all small aircraft gather at X1 and all other arriving aircraft gather at X2. Given that the first crossing group consists of two small (s) aircraft (at the first cross-point X1) and one heavy (h) aircraft (at the other cross-point X2), the crossings at the two cross-points will occur in parallel. The first crossing aircraft is assumed to occupy the runway for 40 sec and each aircraft behind it for an additional 10 sec. Therefore, the first group of crossings will occupy the departure runway for $40 + 10 = 50$ sec (the two small aircraft crossing in simultaneously with the one heavy). This brings us to time point 900 ($= 850 + 50$), which is the end of the crossing operations and the first time that the next available departure can be cleared for takeoff. Based on the above schedule, the next available departure is a Large aircraft that will need 60 additional seconds behind it before another departure can be performed. Therefore, in this case, the next time an operation can be allowed is at time point 960 and the slot midpoint is at $(900 + 960) / 2 = 930$. The remainder of the slot midpoint times can be calculated in a similar fashion. Note that, the calculations are very similar

even if the time assigned to each runway event is set equal to the start (as opposed to the midpoint) of the time slot that the specified aircraft is assigned to.

In a more general case, where an active runway serves all types of operations and in which modifications to the arrival schedule are also permitted, let the original arrival (touchdown) times be T_{on_i} , the actual crossing times of those arrivals be T_{x_i} and the target departure (clearance to takeoff) times be the calculated class slot midpoint values T_{off_j} (as described in the example of the previous paragraph). For every arrival i , $1 \leq i \leq N_A$, where N_A is the total number of arrivals considered and for every departure j , $1 \leq j \leq N_D$, where N_D is the total number of departures considered. $N_A + N_D = N$, is the total number of “mixed” operations on the runway during the current scheduling window. If only departures and crossings are serviced on the runway, then $N_A = 0$.

The delay for each operation is defined as the difference between actual touchdown, crossing or takeoff time (T_{on_i} , T_{x_i} and T_{off_i}) and the corresponding earliest possible values for each flight E_{on_i} , E_{x_i} and E_{off_i} (all times are measured in seconds from time zero). The latter are calculated using the input arrival and departure schedules and average unimpeded taxi out time values for departures, or estimated runway occupancy times for arrivals. Hence, the delay value for each operation represents how much later than its earliest possible time an operation will occur. Then, minimization of the cumulative delay for the runway (i.e. arrival, departure and crossing delay) can be formulated as:

Min aggregate delay:

min (Departure Delay + Arrival Delay + Crossing Delay):

$$\min \left(\sum_{i=1}^{N_D} |Toff_i(p_{TOi}) - Eoff_i|^{k_{D_i}} + \sum_{j=1}^{N_A} |Ton_j - Eon_j|^{k_{A_j}} + \sum_{m=1}^{N_A} |Tx_m - Ex_m|^{k_{X_m}} \right) \quad (3.54) \text{ or}$$

Minimize ONLY departure delays:

$$\min \sum_{i=1}^{N_D} |Toff_i(p_{TOi}) - Eoff_i|^{k_{D_i}} \quad (3.55),$$

where p_{TOi} is the takeoff slot position of aircraft i . Also, k_A , k_D and k_X are positive, flight-specific problem parameters with the following value range: $k_A \geq 1$, $k_D \geq 1$ and $k_X \geq 1$. Giving to any of these parameters a large positive value will penalize delays of specific flights and reduce the amount of delay that a flight will have to suffer. The objective function is generally a non-linear one, unless all k parameters are set equal to one.

Note that, in the context of the simulation results presented later in Chapter 6 (section D.2.2), there will be a few other objective functions introduced, which attempt to not only optimize the operational performance of airport ground traffic, but also take into consideration the amount of workload that this optimization incurs on air traffic controllers as they try to implement the runway operations plans suggested by the optimization algorithm.

3.8.3 The Departure Aircraft Scheduler

The core of the second stage is the Departure Aircraft Scheduler. It generates a family of

Aircraft Schedules each of which corresponds to a different way of populating the class slots of the “Target Class Sequence” (TCS – i.e. the best in terms of throughput class sequence from the first stage) with specific aircraft. For each class slot in the TCS, a Branch & Bound algorithm samples aircraft from the pool of available departing aircraft of that same weight class, in an effort to generate departure aircraft schedules, which satisfy the remaining system constraints that were not addressed in the first stage. Some of the generated aircraft schedules may be unacceptable if they violate system “hard” (inviolable) constraints²¹, such as ATC restrictions. If the first (optimal) schedule is not feasible, then the next available aircraft schedule is chosen (feedback A in Figure 3.2) and checked for feasibility. If all aircraft schedules are exhausted and none is feasible, the Target Class Sequence is replaced with the next available class sequence from the CS matrix (feedback B in Figure 3.2).

3.8.4 Output

The output of the second stage is a matrix AS of a number of aircraft schedules (denoted as n in equation (3.56)) that are listed in ascending objective value order, i.e. each row is an aircraft schedule and the row number reflects the ranking of the specific schedule relative to the other schedules. Thus, the best schedule is listed in the first row and the matrix AS takes the form:

$$\begin{aligned}
 \text{AS} = & \quad [\text{Aircraft Schedule 1,} \\
 & \quad \dots\dots\dots \\
 & \quad \text{Aircraft Schedule j,} \\
 & \quad \dots\dots\dots \\
 & \quad \text{Aircraft Schedule n}] \quad (3.56)
 \end{aligned}$$

²¹ For a definition of “hard” constraints, see [3], [4]

Chapter 4

METHODOLOGY FOR GENERATING

GATE PUSHBACK PLANS

FROM RUNWAY OPERATIONS PLANS

The main objective of the two-stage algorithm for optimizing runway operations plans is to improve the efficiency of the airport's ground operations. However, for controllers to fully take advantage of the algorithm's output, it must be "translated" from an optimal runway operations plan into an optimal gate pushback plan. In this chapter, a methodology for performing this "translation" is presented. The discussion of this methodology will assist in showing in Chapter 6 how the generated optimal pushback plans were useful in the analysis of potential airport performance benefits emanating from using the two-stage algorithm.

4.1 The Significance of Surface Movement Uncertainty in Runway Operations Planning

As mentioned in Chapter 1, there is significant uncertainty in the movement of aircraft and other vehicles on the airport surface. Most importantly, the stochastic behavior of the time that each particular aircraft pushes back from its gate translates into uncertainty in the time that it reaches the runway and becomes ready for take off. This uncertainty has an adverse effect on the airport departure throughput level, because it increases the possibility of “idle” gaps appearing in the sequence of operations on the departure runway and hence, assuming that there is enough departure demand at the gates, it jeopardizes the ability to maintain a certain level of pressure on the departure runway. Consequently, because surface operations planning is performed with the objective of maximizing airport throughput, uncertainty in airport surface movements can also affect the effectiveness of surface operations planning. The following example is intended to clarify the implications of uncertainty in generating successful runway operations plans.

Assume that we have a set of three aircraft A, B and C, planned to take off in that order, with their planned takeoff times at 6:20, 6:21 and 6:22 am respectively. Assume also that the average time to taxi unimpeded from their gates to the runway is 10 minutes for each of the three aircraft. Ideally, in a deterministic airport surface environment, we would want aircraft A to push back from its gate at 6:10 am and arrive at the runway at 6:20 am in time to takeoff. Similarly, we would want aircraft B to push back at 6:11 and aircraft C at 6:12 am. So, if the initial estimates of the planned pushback times for the three aircraft are 6:10, 6:11 and 6:12 am and if all

variables take on values equal to their mean values, then all aircraft will depart as planned and there will be no queuing for any aircraft. However, due to the stochastic nature of gate pushback and taxi times, this almost never happens and the chance of aircraft pushing back from their gates earlier or later as well as the chance of aircraft arriving at the runway earlier (taxi faster than expected) or later (taxi slower than expected) increase the likelihood of “runway-idle” gaps appearing in takeoff operations and departure throughput being compromised.

For example, if aircraft C has a non-zero probability of pushing back late and / or taking a longer than average time to taxi out, then it might arrive at the runway later than planned (e.g. at 6:23 am) and miss its planned takeoff slot by one minute. If the preceding aircraft B takes off on time or earlier, then the late arrival of aircraft C results in an idle gap of at least one-minute at the runway. In such a case, the risk of having a gap in the utilization of the runway may be reduced by planning the pushback time for aircraft C earlier than the initial estimate so that it will almost always be there for its planned takeoff time at 6:22 am.

This example case above is shown in Figure 4.1. The top of the figure corresponds to the situation before planning the pushback time of aircraft C earlier than the initial estimate. The planned takeoff slots for aircraft B and C and their midpoints are shown. In addition, due to the stochastic nature of pushback and taxi operations, each slot is characterized by a probability distribution for the time of arrival at the runway of the corresponding aircraft (B or C). Prior to advancing the planned pushback time of aircraft C, the overlapping region between the two distributions represents the probability of NOT having any time that the runway is not utilized between aircraft B and aircraft C, i.e. the probability of a runway idle gap NOT appearing. The

bottom of Figure 4.1 corresponds to the situation after the pushback time of aircraft C is planned earlier than the initial estimate. Advancing this planned pushback time for aircraft C (while planning for aircraft B remained unchanged) increases the overlapping region between the two probability distributions. Therefore, the probability of a runway idle gap NOT appearing between aircraft B and C is increased, or, in other words, the risk of having a gap in the utilization of the runway is reduced.

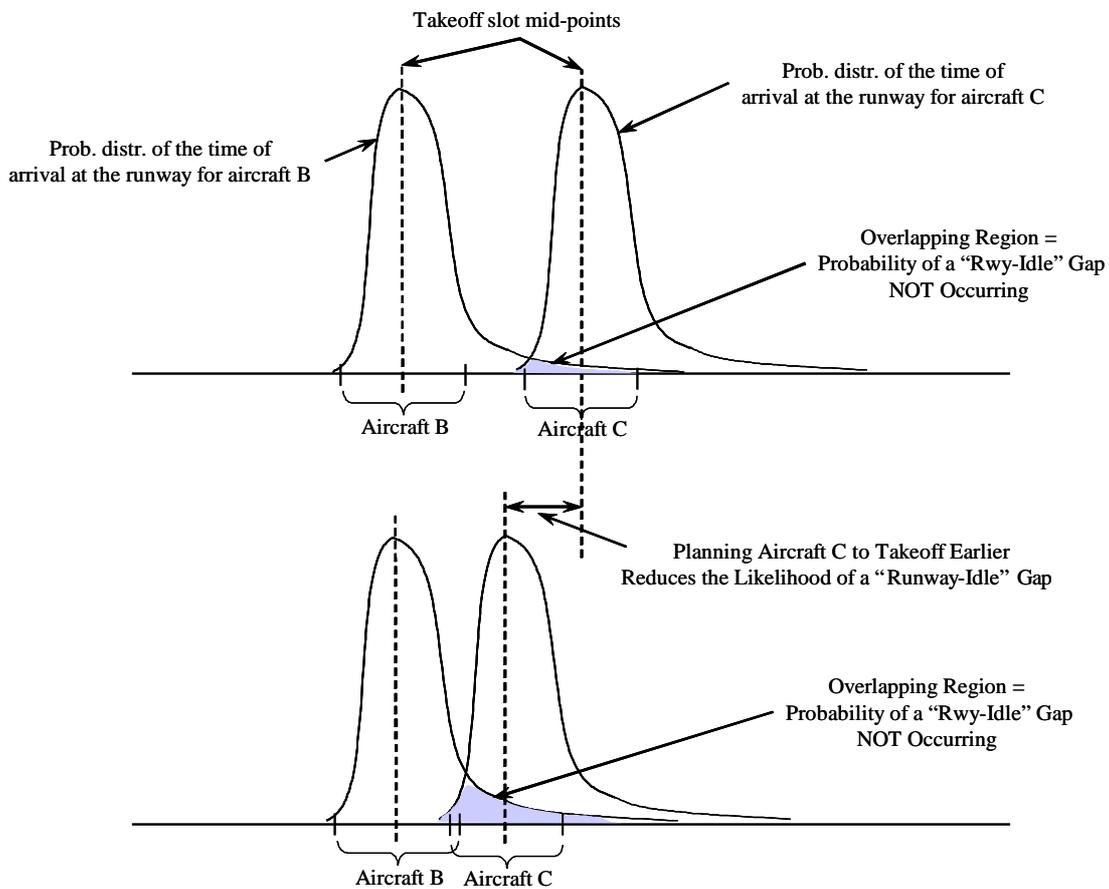


Figure 4.1: Example of Reducing the Likelihood of a “Runway-Idle” Gap by Adjusting Earlier the Planned Pushback Time of the Trailing Aircraft

This advancing of the pushback time for aircraft C is also beneficial in two other cases:

- a) If aircraft B pushes back early and / or takes a shorter time to taxi out then, there is a non-zero probability of aircraft B arriving at the runway earlier than planned (e.g. at 6:20 am). In such a case, if aircraft C is also early (e.g. arrives at the runway at 6:21 am even though it is planned to take off at 6:22 am), then aircraft C can take off as soon as it arrives at the runway and therefore, fill the gap left when aircraft B takes off earlier.
- b) If aircraft A has a non-zero probability of pushing back early and / or taking a shorter than average time to taxi out, but aircraft B arrives at the runway late, then, if aircraft C arrives at the runway early, it can swap places with aircraft B and prevent a runway gap from appearing between aircraft A and B.

As it will be further explained in Chapter 5, the two-stage algorithm takes as an input the originally published pushback schedule. Then the algorithm propagates this original pushback schedule forward to the runway based only on average unimpeded taxi times for each aircraft involved in the planning. By using unimpeded taxi times for the forward propagation, the uncertainty in surface movements is not considered. Therefore, in order to account for movement uncertainty, which is a very important element of the planning process as it was explained earlier in this section, we incorporate the effects of uncertainty in the process that back-propagates (“translates”) the runway operations plans generated by the two-stage algorithm to pushback plans at the gates²². If average unimpeded taxi times are used to back-propagate takeoff times to pushback times, just as they were used in the forward propagation, then the final pushback plan (end product of the back-propagation) will be unrealistic because the effect of

²² ... which is what is finally provided to the controllers ...

surface movement uncertainty is completely ignored. For that reason, the amount of time that back-propagates each aircraft from its planned takeoff time to its planned pushback time has to include:

- a) The average unimpeded taxi time for each particular aircraft, which would be included anyway, and
- b) An amount of pushback time “adjustment” that corresponds to the effect of uncertainty on the ability to exactly materialize the runway operations plans.

In summary, given the two competing "forces" which act on each departing aircraft in the system and cause them to arrive at the runway either early or late, it is necessary that, in the interest of maintaining enough pressure on the runway to ensure a high level of departure throughput, the planned pushback times of aircraft be adjusted to times earlier than those times resulting from back-propagating the planned takeoff times to planned pushback times based only on average unimpeded times. The right balance between these two “forces” and therefore the right amount of pushback time “adjustment” must be determined individually for each aircraft n , given all the possible pushback and taxi behaviors that could manifest for aircraft 1 through $n-1$ that come before it.

4.2 Determining the Appropriate Back-Propagation

Adjustment

As the value of n increases and an aircraft n is further down in the pushback process, it becomes

progressively less likely that all the aircraft in front of n will push back from their gates and taxi early, because the probability that all aircraft preceding aircraft n are early is reduced. Therefore, it seems that the level of pushback time adjustment for each aircraft should be different. As the value of n increases, the time adjustment should also increase, i.e. the adjustment for aircraft n will always be higher than that for aircraft $n-1$. That is because the pushback time adjustment of each aircraft n in the process consists of the following two elements:

- i) The amount of adjustment that was assigned to the previous aircraft $n-1$, which is the minimum that the pushback time of the next aircraft n has to be adjusted in order to avoid creating “artificial” gaps in the takeoff process and
- ii) An additional amount of adjustment to account for the uncertainty in the time of arrival of aircraft n at the runway.

So, for example, in a group of 15 aircraft, the pushback time adjustment for aircraft 10 will be higher than that for aircraft 9 and the latter will be higher than that for aircraft 8.

Also, as the value of n increases, the differential between the adjustment for aircraft $n-1$ and the adjustment for aircraft n is reduced, i.e. in the above example group of 15 aircraft, the difference in adjustment between aircraft 9 and 10 will be lower than the difference in adjustment between aircraft 8 and 9. Using some of the notation from Figure 4.1 and focusing on the first three aircraft of the pushback process, Figure 4.2 explains this decreasing trend of the differential between consecutive pushback time adjustments. The following arguments can be extended similarly for the entire pushback process.

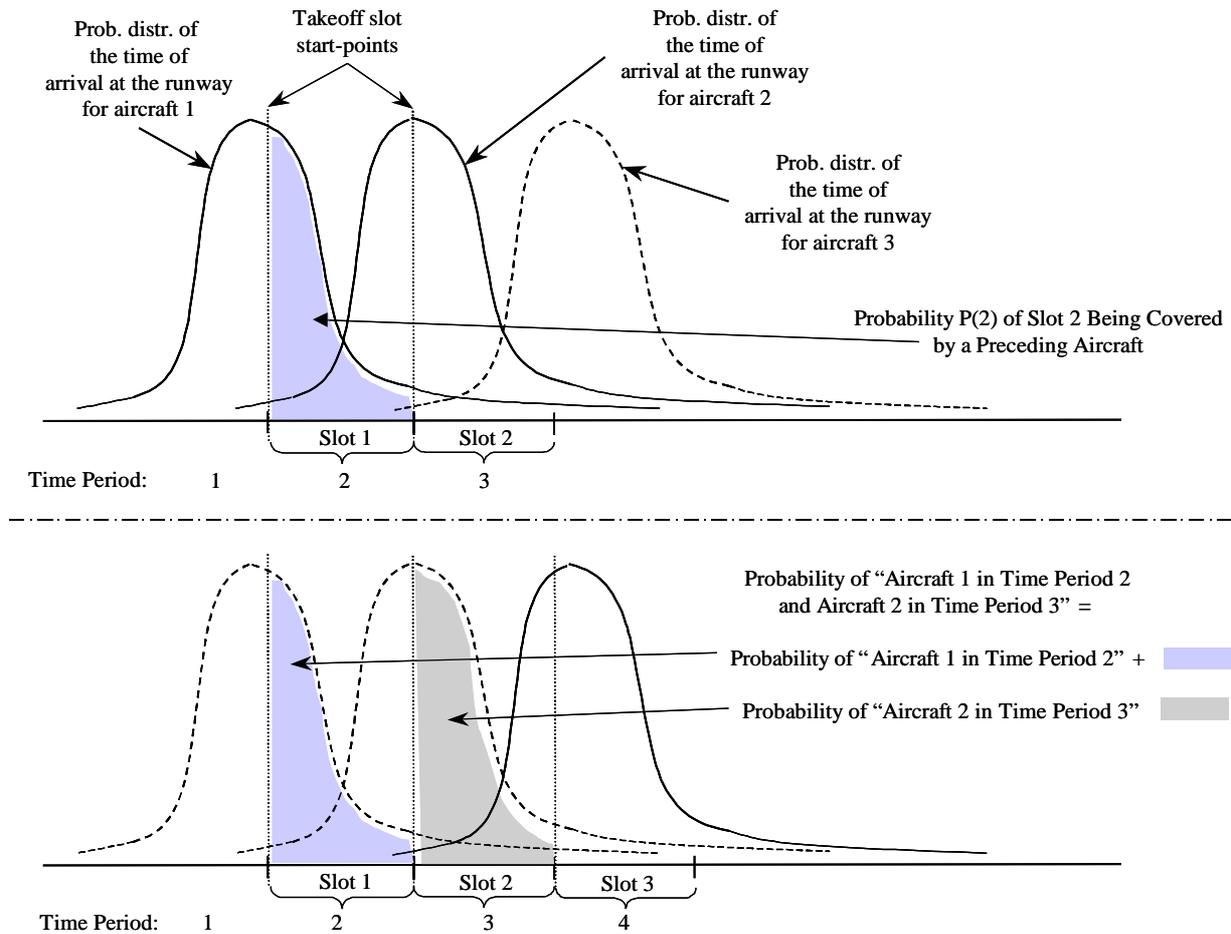


Figure 4.2: Analyzing the Trend of the Differential in Pushback Time “Adjustment”

We assume that, in order for a slot to be covered by the aircraft assigned to it in the runway operations plan that aircraft has to arrive at the runway before the start-point of the slot. So, at the top of Figure 4.2, if aircraft 2, which is assigned to occupy slot 2 according to the plan, is late, i.e. arrives at the runway after the start point of slot 2 and in time period 3, then the probability of takeoff slot 2 being covered by a preceding aircraft, i.e. aircraft 1, is the probability of aircraft 1 missing its assigned slot 1 and arriving at the runway within the next time period 2. We call this probability $P(2)$.

The bottom of Figure 4.2 goes one step further along the process and focuses on the probability of the event that slot 3 is covered by a preceding aircraft, if aircraft 3 is late for covering slot 3. There are several disjoint events that can result in that event depending on when each of the preceding aircraft (1 or 2) actually arrives at the runway, i.e. within which of the time periods 1 through 4 depicted at the bottom of Figure 4.2. Table 4.1 presents all these possible events.

Aircraft 1 arrives in Time Period	Aircraft 2 arrives in Time Period	Slot 3 Covered by Preceding Aircraft ?	Slot 3 Covered By Aircraft
1	1	NO	
	2	NO	
	3	YES	2
	4	NO	
	1	NO	
	2	YES	1 or 2
	3	YES	2
	4	NO	
3	1	YES	1
	2	YES	1
	3	YES	1 or 2
	4	YES	1
4	1	NO	
	2	NO	
	3	YES	2
	4	NO	

Table 4.1: Disjoint Events that can lead to Slot 3 Being Covered by a Preceding Aircraft

The event “aircraft 1 arriving at the runway within time period 2” with probability $P(2)$ that was described at the top of Figure 4.2 is a subset of the event “aircraft 1 arriving at the runway within time period 2 and aircraft 2 arriving at the runway within time period 3”. The probability of the latter is shown at the bottom of Figure 4.2 and corresponds to the highlighted row of Table 4.1. In addition, because all the events presented in Table 4.1 are disjoint, the total probability of the event “slot 3 covered by a preceding aircraft” is the sum of the probabilities of all these events. Therefore, we can easily conclude that:

$$P(3) = P(\text{Slot 3 being covered by a preceding aircraft}) > P(2)$$

This means that slot 3 has a higher probability of being covered by a preceding aircraft in case aircraft 3 is late, than the probability of slot 2 being covered by a preceding aircraft in case aircraft 2 is late. Based on this, element (ii)²³ for aircraft 2 has to be higher than element (ii) for aircraft $n = 3$, because slot 2 has a lower probability of being covered by a preceding aircraft (in case aircraft 2 is late) than the probability of slot 3 being covered by a preceding aircraft. This is why, as the value of n increases, the differential between the adjustment for aircraft $n-1$ and the adjustment for aircraft n is reduced.

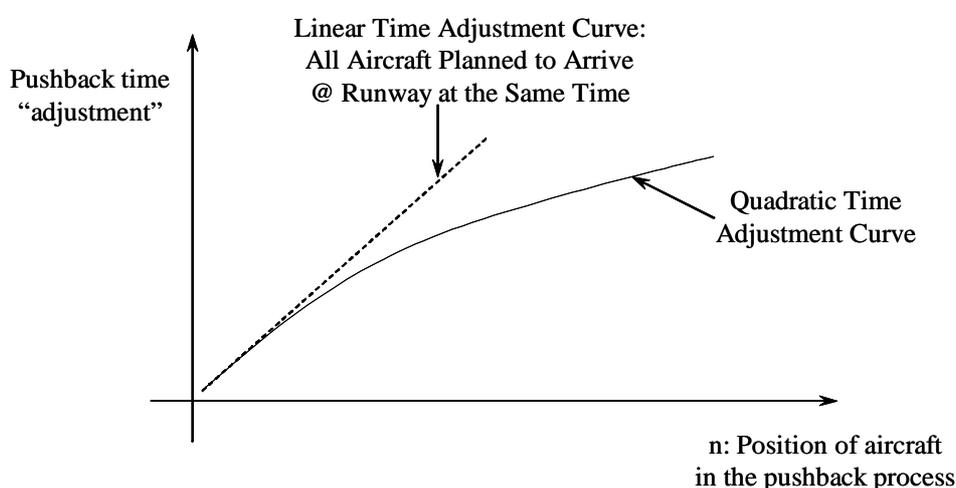


Figure 4.3: Suggested Quadratic Trend for the Pushback Time “Adjustment”

Figure 4.3 is an empirical diagram that gives an idea of the trend described in the two points above, which suggests a quadratic behavior. In the extreme case where all aircraft in the process are planned to arrive at the runway at the same time, each aircraft n in the process takes the same amount of additional time adjustment as the one that the preceding aircraft $n-1$ took, i.e. element

²³ ... among the two elements mentioned at the beginning of this section which comprise the pushback time adjustment of each aircraft in the process ...

(ii) of the time adjustment for each aircraft has always the same value (as opposed to the decreasing value that was described earlier). This means that the quadratic curve in Figure 4.3 becomes a linear curve.

In summary, the decreasing trend of the differential in pushback time adjustment between aircraft $n-1$ and aircraft n can be explained by the fact that, the few aircraft at the front end of the pushback process are the ones that do not have a lot of other aircraft ahead of them. Therefore, if they are late to cover their assigned takeoff slots, the resulting runway gap is less likely to be countered for by the uncertainty in the movement of other aircraft ahead of them. Understandably, because throughput values are heavily affected by idle gaps that can appear at the runway due to ground operations uncertainty, the pushback times of the first few aircraft are the main “drivers” (determinants) of the final departure throughput. On the other hand, aircraft that are later in the pushback process have a larger number of aircraft ahead of them, which collectively bring to the system a higher level of uncertainty, enough in most cases to counter for possible runway idle gaps resulting from aircraft being late for their takeoff slots and reduce the likelihood of a lowered departure throughput.

At this point it should be mentioned that the above arguments are valid only in a phase during which the flow of departing aircraft from the gates to the runway is still building up. This can occur if aircraft pushbacks are scheduled in clearly separated “banks”. On the other hand, once the aircraft flow has settled to a constant level and there is always a continuous outflow of aircraft from the gates to the runway, then, for each aircraft in the flow there are always enough aircraft in front of it and behind it, which makes this situation different from the one described

above.

In an effort to obtain a heuristic curve for the back-propagation time adjustment to be used in the simulations as a function of the value of n , i.e. as a function of the position of each aircraft in the pushback and taxi process, we observed the queuing process in Boston's Logan airport (Figure 4.4 based on data from [74]).

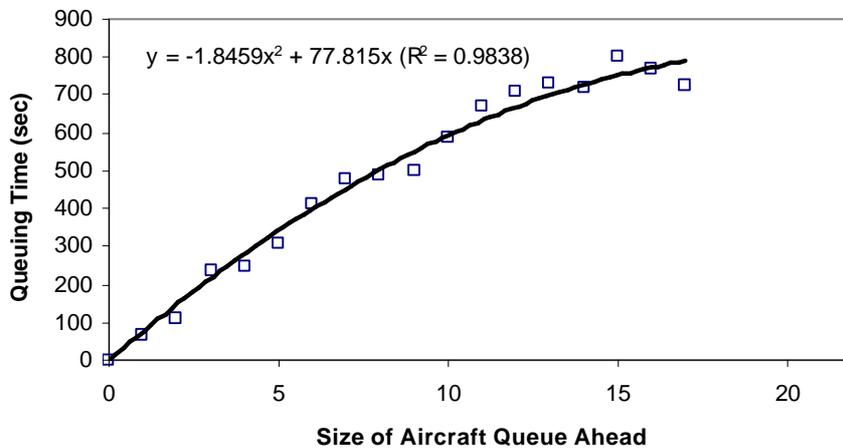


Figure 4.4: Logan Configuration 22L / 22R / 27 – Queuing Time Behavior

Data was collected and used to analyze the queuing behavior under a particular runway configuration (22L / 22R / 27) [74]²⁴. It was discovered that the amount of time that aircraft spent at the runway queue, i.e. the time between entering the runway queue and taking off, may be modeled well by a quadratic form as a function of the size of the queue that an aircraft experiences ahead of it (and therefore as a function of the position of the aircraft in the queue).

²⁴ This is also the Logan runway configuration that is modeled in the benchmark airport presented in Chapter 5 (Figure 5.1).

The main reason for this quadratic behavior is the fact that, as the size of the aircraft queue increases, each aircraft that follows, reaches at the queue earlier (because the aircraft queue extends to a larger length) and therefore, becomes progressively more likely to enter the queue early. This means that, on one hand, the last aircraft to enter the queue spends more time waiting in it than any other aircraft ahead of it, but on the other hand, it does not absorb an amount of queuing time that is linearly dependent on the size of the queue. If the queuing behavior was linear it would be as if each aircraft pushes back from its gate early enough such that it arrives at the runway unimpeded before any of the aircraft ahead in the queue takes off and then watches the aircraft ahead in the queue take off before it is its turn to takeoff.

Let us see an example of the above behavior: assume that aircraft D is scheduled to take off at 6:20 am and aircraft A, B and C are scheduled to takeoff at 6:12 am, 6:18 am and 6:19 am. Also, aircraft D has an unimpeded taxi time of 10 minutes and therefore, its tentative pushback time is 6:10 am. If a fixed amount of queuing adjustment was used for each aircraft ahead in the queue, let's say 50 sec, then, given that there are three aircraft in the queue ahead of D (aircraft A, B and C), the queuing adjustment of the planned pushback time for aircraft D would be $3 \times 50 \text{ sec} = 150 \text{ sec}$ and its final planned pushback time would be set at 6:07:30 am. Then, aircraft D would arrive unimpeded at the runway at 6:17:30 am and would watch aircraft A takeoff while taxiing out (6:12 am) and aircraft B and C takeoff while waiting in the runway queue (6:18 and 6:19 am). However, the impact of aircraft A scheduled to take off at 6:12 am on aircraft D is minimal and definitely much less expected than the impact of the two aircraft B and C scheduled to take off at 6:18 and 6:19 am. That is why the queuing behavior has a quadratic form, which almost corresponds to multiplying a fixed amount of queuing adjustment (i.e. the 50 sec used earlier as

an example) by the probability of the impact of the aircraft ahead in the queue. For any given aircraft n , this probability of being affected by the aircraft ahead of it in the queue is reduced as a function of the position of aircraft n itself in the queue.

In summary, it was determined that, when calculating a planned pushback time from a planned take off time for each aircraft in order to generate a realistic pushback plan, an amount of queuing time adjustment needs to be subtracted in addition to subtracting the average unimpeded taxi time. Given the similarity between the observed queuing behavior and the behavior in the process that was described earlier in this section, which drives the probability of aircraft needing a larger or smaller pushback time adjustment, it was decided (as we will also see in the simulation results in Chapter 6) that the amount of pushback time adjustment to be used will be heuristically calculated based on the estimated queue size that each aircraft is expected to encounter ahead of it. This will reproduce the suggested quadratic trend and will assign a different value of pushback time adjustment to each aircraft in the pushback process, as the number of aircraft expected to be ahead of it changes.

It must be mentioned that, if the initial estimate of the planned pushback time is adjusted by the amount given in that queuing data set from [74], from which the queuing time behavior of Figure 4.4 was derived, it is as if we basically recreate the amount of queuing that existed in the traffic situation on the day that the data was collected. In order to work around that limiting factor, additional values for the coefficients of the quadratic queuing curve must be tested²⁵ to

²⁵ The exact method used for testing additional curve coefficients is described in section 6.3.2.

investigate if it is possible to determine a quadratic curve that is "better" than the one in Figure 4.4 (which corresponds only to the application example for the particular Logan runway configuration 22L / 22R / 27). The term "better" is used here in the sense that the new curve would yield lower average values for the time-based performance metrics (cumulative delays and taxi times) without compromising the achieved average throughput values. The latter is a concern, because, when shifting the quadratic curve used (Figure 4.4) up or down, it is easy to "fall victim" to the existing tradeoff between system throughput and system delays. To understand this tradeoff, we consider the following two possible extreme cases:

- a) Using zero time adjustment, which would probably result in smaller taxi times and queuing delays but would also cause takeoff slots to be lost at the runway (and therefore reduce throughput performance) and
- b) Using a high (constant for every aircraft) adjustment value, e.g. $2 \times (\text{taxi time standard deviation for each aircraft})$, which would be too large resulting in excessive taxi times and queuing delays but would plan all aircraft to push back early enough so that the risk of idle runway gaps is minimized and therefore throughput performance is increased.

This "tradeoff" leads to the fundamental research question of how big a queue should be maintained at the runway in order to successfully balance between runway throughput and taxi / queuing delays.

Chapter 5

SIMULATION TEST BED

The performance behavior of the proposed “two-stage” Runway Operations Planning algorithm was evaluated with simulation tests implemented using Matlab and Simulink. The rationale for choosing Matlab is that it offers a robust programming environment, which also supports data structures, and the combination of Matlab and Simulink offers rapid prototyping capabilities and modularity. Thus, it is easy to make changes to the airport surface and runway geometry being modeled.

The three main goals of implementing the proposed algorithm through simulation were to:

- a) Determine potential benefits across a specific dimension (e.g. runway throughput or taxi-out delays) that may result from using the proposed algorithm for planning ground operations compared to an example “naïve” case where no sophisticated planning of ground operations is performed by the controllers.
- b) Test the “stability of plans” generated when using the proposed algorithm, by investigating

how much the output plans change when the simulation is fed with stochastically varying inputs and

- c) Investigate the “strength” of the chosen pivot element to act as the basis of the decomposition by examining the robustness of the algorithm output in terms of throughput and delays.

5.1 Benchmark Airport

The Matlab / Simulink model presented here is based on the benchmark example airport shown in Figure 5.1, which describes a runway geometry that is frequently encountered at airports.

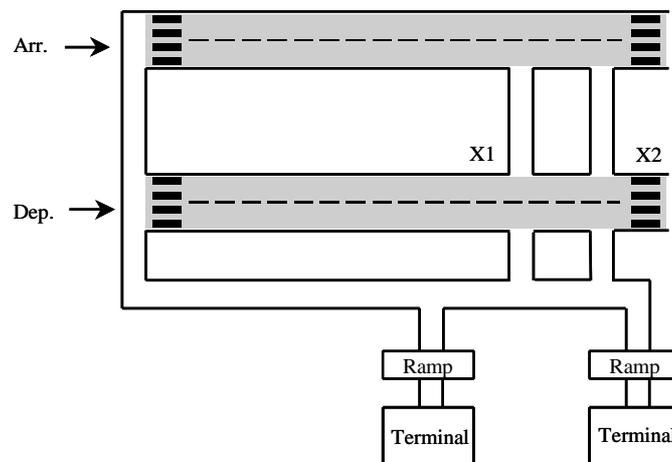


Figure 5.1: Benchmark example airport system

This hypothetical airport system has two parallel runways, one dedicated to arrivals and the other to departures. Aircraft landing on the arrival runway eventually gather at one of the two cross-points X1 or X2 and wait for clearance to cross the departure runway on the way to their gates.

Departures also interact with arrivals on the taxiways, ramps and gate areas, as they taxi to the departure runway for takeoff. Given the departure aircraft classes in hand and the cross-point taxiway capacities, the first stage of the solution procedure is to design the best departure sequence i.e. the departure sequence that maximizes runway utilization (throughput).

Note that, in the simulation, randomness associated with the touchdown times of aircraft in the arrival stream is ignored and arrivals are treated as fixed requests for runway time. Also, the runway occupancy times for departing and arriving flights are assumed constant but different for each aircraft weight class category. Randomness that is encountered in all other aspects of surface operations planning, such as gate pushback and taxi out operations are considered in the simulation model described in the following sections.

5.2 Overview of the Simulation Model

An overview of the Simulink test model in its current state is presented in Figure 5.2. A complete list of model inputs and outputs is provided in Page 27. The model input is in the form of schedule files that contain arrival and departure data structures. These data structures list the details of the flights that are scheduled to operate in and out of the airport, such as: terminal the aircraft is coming from (important for taxi time calculations), aircraft type and class, destination of a departing flight or expected landing (touchdown) time of an arriving flight.

Each field contains characteristics of a departing or arriving flight, which are essential to the

model. The schedule files have a modular structure that offers the ability to add new characteristics if necessary, depending on how the design and the demands of the model evolve. In real-world applications, these schedule files can potentially be populated automatically through a schedule feed probably from the ATC Host computer.

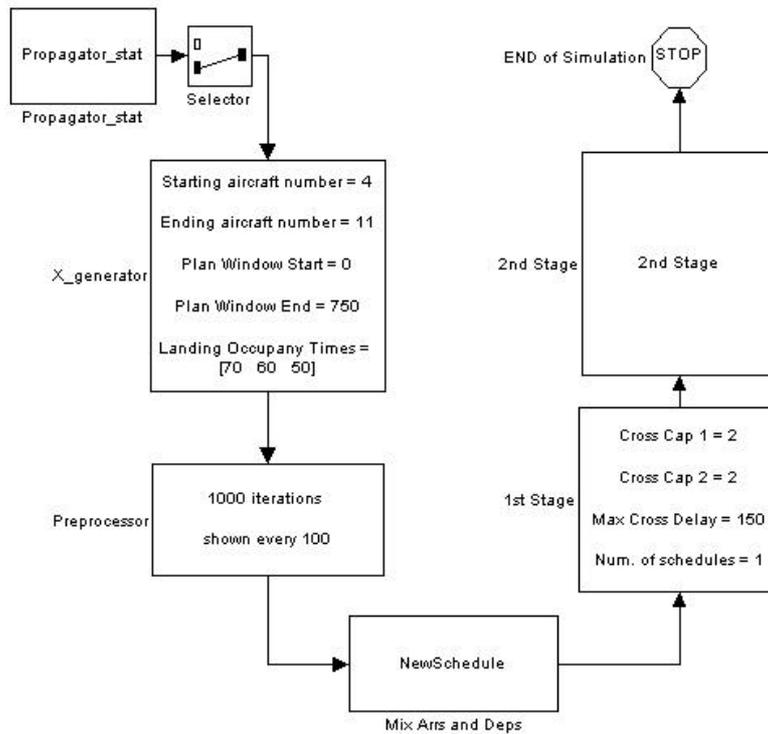


Figure 5.2: Simulink Model Overview

5.3 Model Components

5.3.1 “Gate to Runway” Aircraft Propagation Times

The first model block in Figure 5.2 is called “Propagator” and it can be seen in more detail in

Figure 5.3, representing the example airport of Figure 5.1, which will also be our base example.

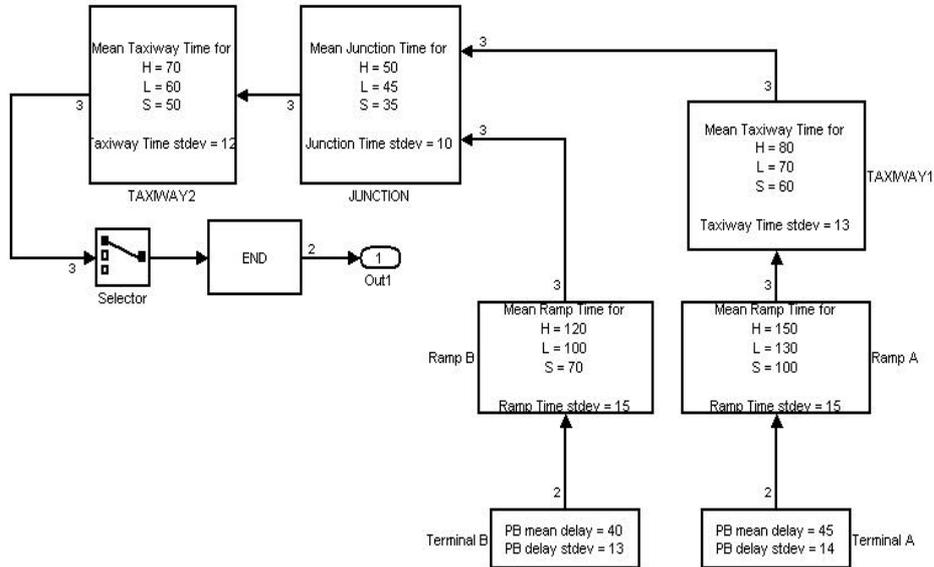


Figure 5.3: Propagator

Its task is to propagate “dummy” aircraft through the airport model in order to develop a “propagation” file with the minimum (unimpeded) times (mean and standard deviation values) necessary for each aircraft type to move from its gate to its assigned runway. Each dummy aircraft is virtually moved through the entire airport model based on the propagation times (time to traverse) that appear in the model blocks representing the airport components. The three numbers in each block are the propagation values corresponding to each of the three aircraft weight classes involved in the problem (Heavy, Large and Small). For example, a large aircraft will need at least 130 time units to go through Ramp A, but will only need 100 time units to go through Ramp B, possibly due to the difference in gate layout between Ramps A and B.

The Propagator is designed based on a library of standard model blocks, such as the Terminal, Ramp, Taxiway and Junction. This provides modularity so that any airport layout can be modeled quickly, by connecting appropriately the necessary standard model blocks. For example, by simple inspection of Figure 5.3, it can be seen that, if the airport had a third terminal / ramp area connecting to the system after the existing Junction and onto Taxiway2, the model could be easily modified by adding one more Terminal and one more Ramp block and finally linking them to the system through an additional Junction block downstream from Taxiway2.

The following three methods for deriving aircraft propagation values (mean and standard deviation of taxi times from gate to runway) were considered:

- *Method 1:* Analyzing approximately four (4) hours of detailed ground traffic data obtained through observations at the Boston Logan airport control tower on February 2, 1999 [74]. High level of detail is required in order to be able to determine individual taxi times for each ramp and taxiway segment on the airport surface
- *Method 2:* Applying the prototype taxi model introduced in [75] and implemented in [30] on the appropriate Airline Service Quality Performance (ASQP) data set (Logan airport departures under specific configurations) can help to predict the unimpeded taxi time from the gate to the runway for the particular configuration and
- *Method 3:* Analyzing ASQP data from nighttime operations (when traffic levels are lower and therefore most taxi operations are unimpeded by traffic) can help estimate the unimpeded taxi time from the gate to the runway.

Each of these methods has its limitations and drawbacks. In order to achieve statistical

significance in the results of the first method, an extensive data set is required, which is difficult to collect. The second and third methods do not provide taxi time information for individual ramp and taxiway segments and at most airports, the lack of information about runway configuration and gate used by each aircraft limits the fidelity of the results that are obtained. For the purposes of this research, in order to generate unimpeded taxi times, method 3 was used in combination with:

- a) Historical runway configuration data provided by the Logan Noise Research Office and
- b) Our knowledge about the location of specific gates and their utilization by particular airlines and aircraft types.

5.3.2 Generating Crossing Requests

The “X_generator” block in Figure 5.2 scans the arrival schedule and based on the expected touchdown time of each arriving aircraft and the expected runway occupancy times (depending on weight class), it calculates the time at which each arrival is expected to become a crossing request. Within the scope of this work, randomness in arrival operations (touchdown times and runway occupancy times) is ignored for simplicity. However, considering randomness in the arrivals stream creates an interesting problem for future research.

5.3.3 Generating Possible Class Sequences

The planning window is set to about 15 to 20 minutes into the future. It is impractical to plan

ahead for a longer period of time, because the length of the planning window is anyway limited by the uncertainty inherent in the system. Besides, 15 to 20 minutes is typically the “plan-ahead” window that most air traffic controllers use, and at the same time it is slightly longer than what it takes for most aircraft (on average) to taxi from their gates to the runway. The aircraft scheduled to push back within the planning window create the pool of available aircraft, each of which has a weight class associated with it. The “Preprocessor” is responsible for developing a list of possible class sequences using the weight classes in hand.

The first stage of the ROP algorithm determines the best class sequence with respect to a throughput objective. This is achieved by calculating the throughput for each of the class sequences generated by the Preprocessor and then by sorting the list of sequences accordingly. As expected, the number of possible class sequences to consider depends on the number N_D of departures involved. Specifically, if the N_D departures consist of N_{DH} heavy aircraft, N_{DL} large and N_{DS} small aircraft, the number of possible sequences is:

$$\frac{N_D!}{N_{DH}! \cdot N_{DL}! \cdot N_{DS}!}$$

In most practical instances, this number will be very large. For example, assume that there are twenty (20) aircraft within the planning period and that the corresponding weight classes are: three heavies (H), twelve large (L) and five small (S). The effective number of possible class sequences is equal to $20! / (3! \cdot 12! \cdot 5!) = 7,054,320$. For this reason, a different approach was implemented. A pre-specified number of randomly generated sequences are created and then pruned to eliminate repetitions.

For the particular original pushback schedule input of a fixed weight class mix of 13.33 % Heavies, 66.67 % Large and 20 % Small aircraft, which was used in some simulation tests presented in Chapter 6, the number of randomly generated sequences required to adequately explore the solution space was determined, through sensitivity analysis, to be in the order of 5,000. More specifically, the Preprocessor was run several times and each time it was allowed to generate an increasingly higher number of possible class sequences. Also, each time, repeated class sequences were deleted from the set of class sequences that the Preprocessor generated and the output set of unique sequences was stored. Then, all stored sets of unique sequences were compared and after several runs, the following conclusions were drawn:

- a) When the Preprocessor was set to generate originally (i.e. before repetitions were eliminated) MORE THAN 5,000 possible class sequences, the difference in the size of the final set of unique sequences, compared to the case where the Preprocessor generated EXACTLY 5,000 possible class sequences, was only to the order of 4 to 5 sequences,
- b) There were even cases in which the final sets of unique class sequences had the same size whether the amount of sequences originally generated by the Preprocessor was equal or more than 5,000 and
- c) In both of the above two cases (a) and (b), the difference in content (not in size) of the final sets of unique sequences between the cases of “exactly 5,000 original class sequences” and “more than 5,000 original class sequences”, was also very small, namely to the order of 2 to 3 different sequences detected.

Similar analysis was conducted for schedules with a different weight class mix that were used in the simulation tests of Chapter 6. The number of generated class sequences for other schedules was higher or lower than 5,000 depending on the weight class mix in each particular case, which

determines the total number of possible sequences that can be generated. For example, an almost homogeneous schedule of 6.667 % Heavies and 93.33 % Large aircraft has a much smaller total number of possible sequences and therefore the number of generated class sequences in the simulation never reaches 5,000.

In the simulation model, the planning window of the algorithm can be specified either as time-based, i.e. by setting a start and end time, or aircraft-based, i.e. contain a certain group of aircraft from the schedule for which scheduled pushback times are between the start and end time.

5.3.4 Mixing Departures and Crossings

The “NewSchedule” block scans through the departures in the schedule file and propagates each of the departing aircraft from their gates to the time they will become departure requests at their assigned runway. This obtains an initial estimate of the runway time window that these aircraft will occupy. Based on research conclusions from [94], the probability density function for the taxi time is assumed to be a lognormal distribution with mean and standard deviation values equal to those in the file with the taxi time propagation values created by the Propagator block. Then, the NewSchedule function mixes the departure requests and the crossing requests (X_generator) to produce a new schedule for the departure runway. The schedule output of NewSchedule includes all departures within the planning period and all arrivals expected to interact on the departure runway with those departures. It is used as an input for the next model block, the Class_Scheduler.

5.4 Simulation of the First Stage of the Algorithm

5.4.1 Generating Class Sequences

The “1st stage” block in Figure 5.2 serves as a “mask” for three other model blocks. The first is the “Class_Scheduler” block, which contains the function that, based on applying ATC wake vortex separation times between successive departures, calculates the throughput (before time for crossings is included) for each class sequence in the list of departure class sequences that was created during preprocessing.

5.4.2 Introducing Crossings in the Class Sequences

The second “masked” block is the “Do_Crossings” block, which introduces the necessary crossing gaps in these schedules and adjusts the throughput value for each schedule, since the introduced crossing gaps affect departure throughput. The location and length of those crossing gaps depend on the structure of the arrival stream as well as on taxiway capacity and maximum crossing delay constraints, which are input in the “1st stage model” block. For example, in the model of Figure 5.2, there is a taxiway capacity of two (2) units for each crossing point and a maximum crossing delay constraint of 150 time units.

“Smart” Crossings

At this point, the departure class sequences are “enriched” i.e. runway crossings are introduced

into the departure schedule using two heuristics as follows:

- Runway crossings are performed in “groups” i.e. multiple crossings at the same time, based on the assumption that there is arrival pressure and therefore there is always a number of aircraft waiting to depart so that no time slot will go unused. If there is no departure pressure, then crossings are served as they arrive, without the necessity of waiting until a “group” of crossing requests has been generated. Grouping the crossing is desirable because the time required to complete multiple crossings is less than the total time required to cross each aircraft individually.
- Runway crossings are scheduled (as much as possible) after “heavy” departures if there are heavy aircraft in the pool of departing aircraft. This is desirable because the required wake vortex separation time behind a heavy departure is larger than the required wake vortex separation behind an aircraft of any other weight class. For the typical airport geometry, this separation time is equal to or greater than the time required to perform multiple crossings.

Both of these heuristics are subject to constraints on the number of aircraft that may be held at runway crossing points (due to physical taxiway space limitations) and on the maximum ground delay that may be imposed on arriving aircraft.

5.4.3 Calculating the Throughput of Class Sequences

The third “masked” block is the “Throughput” block, which calculates the stochastic throughput for each class sequence with crossings, based on the method presented below. The stochastic nature of ground operations mandates that we take into account the possibility that a class slot

may not materialize in its originally scheduled position in the Target Class Sequence. Air traffic controllers may need to alter the sequence of takeoffs in a way that the weight class category of the aircraft released in a particular slot does not match the weight class characteristic that this slot had in the originally planned schedule. This leaves no choice but to calculate stochastic throughput based on potential slot position shifts using probabilistic distributions for the pushback and taxi processes.

Using as a “base” sequence one of the departure class sequences with crossings, these distributions help determine the probability of a class slot actually being at the position it has in the “base” sequence, as opposed to occupying one position up or down in the sequence (only shifts of one position were examined for simplicity). For each “base” sequence, its final stochastic throughput is calculated as the expected throughput over all the possible sequences that can be derived from the “base” sequence by performing possible class slot shifts.

For example, assume that there are fourteen (14) departures within the predetermined planning window. Also, assume that it has been estimated that these departing aircraft are expected to interact with four (4) arrivals, which will request runway time in order to cross the departure runway. Therefore, a possible departure class sequence including crossings (crossings are denoted again by lowercase letters as it was originally stated in section 3.8.2) is:

L - L - L - H - **s/s/h** - H - **l** - L - L - L - L - L - L - S - S - S

As shown in Figure 5.4, in this sequence, which is considered to be the “base” sequence, there are only three (3) possible departure slot swaps (one-position swaps examined only) that can

actually affect the throughput associated with this sequence, (X1 and X2 are abbreviations for the two crossings groups).

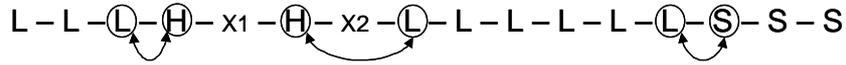


Figure 5.4: Possible class slot swaps

Taking all possible combinations of occurrence of these three swaps, the set of possible class sequences that can be derived from the “base” sequence consists of eight (2^3) sequences (including the “base”), which are shown in Figure 5.5.

- L - L - L - H - x1 - H - x2 - L - L - L - L - L - L - S - S - S
- L - L - L - H - x1 - H - x2 - L - L - L - L - L - L - **S** - L - S - S
- L - L - L - H - x1 - **L** - x2 - **H** - L - L - L - L - L - L - S - S - S
- L - L - L - H - x1 - **L** - x2 - **H** - L - L - L - L - L - L - **S** - L - S - S
- L - L - **H** - L - x1 - H - x2 - L - L - L - L - L - L - L - S - S - S
- L - L - **H** - L - x1 - H - x2 - L - L - L - L - L - L - **S** - L - S - S
- L - L - **H** - L - x1 - **L** - x2 - **H** - L - L - L - L - L - L - S - S - S
- L - L - **H** - L - x1 - **L** - x2 - **H** - L - L - L - L - L - L - **S** - L - S - S

Figure 5.5: Sequences derived from the “base” sequence by performing all possible swaps

For each sequence, the mean value for the starting time-point of the first class slot is the mean “Time at the Runway” for the earliest aircraft in the departure pool that has the same weight class as the starting class slot. In this example, this would be the time at the runway for the earliest large (L) aircraft. The pushback process is assumed to be independent from the remainder of the taxi process and therefore, the “time available at the runway” may be calculated

as the sum of the mean pushback time (including pushback delays) and the mean taxi time (from gate to runway threshold) for each specific aircraft and for the particular terminal it is coming from. Once the time of the class slot for the first departure is determined, the times of the other class slots can be determined from wake vortex separation criteria and the duration of the activity in each slot. Then, we can assign a lognormal distribution curve to each slot using the starting point or middle point of the slot as the "mean value" and the taxi time standard deviation as the standard deviation. The overlapping region between curves of adjacent class slots, determine the probability of a "one-position" swap between those two slots occurring, as shown in Figure 5.6. Based on those swap probabilities and the combination of swaps involved in each derived sequence, a probability of occurrence and a throughput value for that particular derived sequence can be calculated. The final stochastic throughput for the "base" sequence is calculated as the expected throughput over the throughput values of all the derived sequences, each of them considered with its individual probability of occurrence.

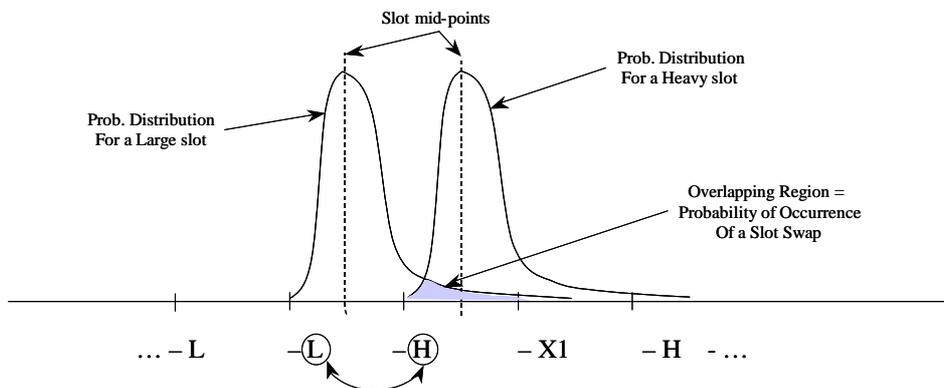


Figure 5.6: Probability of Occurrence of a Slot Swap

This process is repeated for each class sequence with crossings, and finally the list is ordered according to stochastic throughput in descending order. A number of departure class sequences

with crossings (starting at the top of the list) can then be considered the best in terms of throughput and become “Target Class Sequence” candidates in the second stage of the algorithm. The number of Target Class Sequences considered can be adjusted and even the entire list can be examined (as opposed to a “top” subset of the list).

5.5 Simulation of the Second Stage of the Algorithm

5.5.1 Entering Air Traffic Control Flow Restrictions

The “2nd stage” block in Figure 5.2 is a “mask” for two model blocks. During a simulation run, when the time comes for the “2nd stage” block to be executed, the first “masked” component prepares and sequentially displays the two prototype Graphical User Interfaces (GUIs) snapshots of which are shown in Figure 5.7 and Figure 5.8.

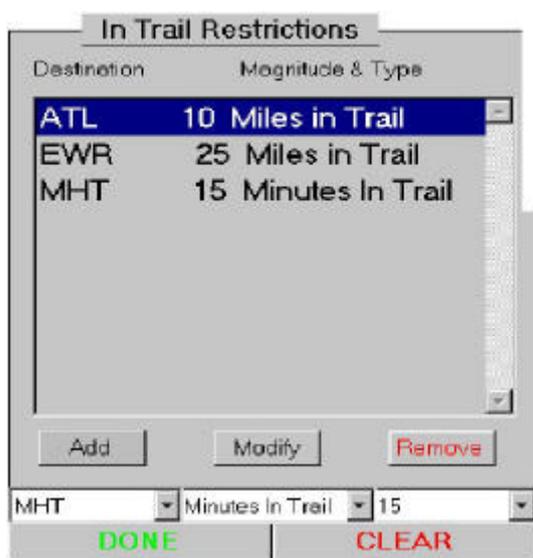


Figure 5.7: GUI for In Trail restrictions



Figure 5.8: GUI for Time Window restrictions

Before optimization functions are deployed, ATC constraints such as Miles (Minutes) In Trail (MIT), Expected Departure Clearance Times (EDCT) and Departure Sequencing Programs (DSP) are entered in the system through these two GUIs.

5.5.2 Branch & Bound Solution

The “Aircraft_Optimizer_X” model block (also “masked” under the “2nd stage” block in Figure 5.2) contains the objective and constraint functions of the Integer Program solved in the second stage of the algorithm, as these were described in Chapter 3.

While the optimization toolbox in Matlab has functions to solve linear programs, there are no built-in functions to solve pure integer programs (in which all the decisions variables can only assume integer values). However, a function that solves integer programs, which was developed by the Department of Industrial Engineering at North Carolina State University [89] was downloaded and implemented. This function uses Matlab’s built-in optimization functions to perform a very efficient Branch & Bound algorithm implementation. Other integer program solvers that use Matlab’s optimization toolbox are also available for download, such as [111]. However, the one in [89] was chosen because it is very fast (see section 5.5.3) due to the fact that it does not involve any recursive calls of the optimization function.

The Branch & Bound algorithm offers a solution method for integer programming or mixed integer programming problems, which is based on an implicit enumerative evaluation of the set

of feasible solutions. The approach uses the “divide and conquer” logic in order to explore the set of feasible integer solutions [18].

5.5.3 2nd Stage Problem Size and Simulation Runtime

The final size of the problem to be solved with the Branch & Bound method depends on two factors:

- a) The number of aircraft involved in the planning, which can take any value, even though, given the uncertain nature of airport surface operations, there is an upper limit to how many aircraft it is practical to plan for and
- b) The number and type of ATC restrictions (EDCT, DSP and Miles (Minutes) in Trail) that are active at any point in time and therefore need to be formulated and included in the problem.

The simulation was run in two different computers, a Dell Dimension XPS T700r desktop with a 733 MHz processor and a Dell Inspiron 8200 Notebook with a 2 GHz processor. Keeping the number of randomly generated class sequences the same in all cases, the runtime for the first stage of the algorithm depended on:

- a) The processing speed of the computer used,
- b) The number of aircraft involved in the planning and
- c) The (pre-selected) number of class sequences for which the stochastic throughput calculation was performed.

For the cases where stochastic throughput calculations (which take more time to be completed) were not performed, the first stage, i.e. the complete random generation of class sequences with crossings and the calculation of the associated throughput values, was completed in anywhere between approximately 4 and 10 seconds for a number of aircraft involved in the planning ranging between 9 and 20. The runtime for the second stage of the algorithm did not vary significantly from one processor to the other for the exact same integer-programming problem. For the majority of the largest cases, i.e. the ones with the largest final problem size (as this was determined by the two factors mentioned at the beginning of this section), and assuming that the problem is a feasible one, it took at most 4 to 5 seconds for the 2nd stage Branch & Bound process to reach an optimal solution. Note that, purely for comparison reasons, the 2nd stage of the simulation was also solved using the solver available at [111]. Due to the recursive nature of the latter, it took (for feasible problems) approximately 7 to 8 seconds (on the Dell Inspiron 8200 Notebook) up to approximately 60 seconds (on the Dell Dimension XPS T700r desktop) for this solver to reach to an optimal solution.

5.6 Solution Examples

In the following solution examples, the problem is decoupled and departures are isolated from arrivals. Exactly as in Figure 5.1, one runway is used for departures and arrivals are landing on a parallel runway. Similar but slightly more complicated numerical examples can be created for cases in which, runways are used for mixed operations (arrivals and departures).

5.6.1 Stage One

Assume that the first six available departures in the “pool” of available aircraft are three small aircraft (S), two large (L) and one heavy (H) and the order they called ready for pushback is: S - S - H - S - L - L. Under a First Come (Call for Pushback) First Serve (Clear to Take Off) control policy, there will be no modifications in the takeoff sequence and timing:

$$S (200) - S (260) - H (320) - S (440) - L (500) - L (560) - N (560 + 60 = \mathbf{620})$$

where:

- The numbers in parentheses are the scheduled takeoff times, given the required separations (in seconds) between each pair of aircraft based on their respective weight classes. The scheduled takeoff time of the first aircraft in the sequence is set to a value greater than zero just as an indication that this is a part of a series of consecutive departure plans along an entire operational day²⁶ and
- N is the class type of the next aircraft taking off right after the last aircraft of this departure group, which will be separated from the last large (L) aircraft by 60 sec whether it is a heavy, a large or a small departure.

It takes a total of 420 seconds for this sequence of six (6) aircraft to complete their takeoffs. However, if the heavy aircraft is positioned last in the takeoff queue, it may lead to throughput savings (or in the worst case no throughput loss) because it requires the largest takeoff separation among all aircraft weight classes (120 sec if followed by a small or large and 90 sec if followed

²⁶ Even if the first scheduled takeoff time had been set equal to zero, the arguments in this example would still be valid.

by another heavy aircraft). In that case, the six aircraft can complete their takeoffs in at most 420 sec, or in 390 sec (if the next aircraft N is a heavy) saving 30 sec of runway time:

S (200) - S (260) - S (320) - L (380) - L (440) - H (500) - N (500 + 90 or 120 = **590 or 620**)

Note that, placing a heavy at the end of the sequence may unnecessarily penalize heavies in the final solution. As the following example will show, it often makes better sense to place a heavy just before a group of runway crossings. If the arrival stream is taken into consideration, arriving aircraft introduce requests to cross the departure runway and therefore a certain amount of time on the departure runway has to be used for these aircraft to cross. It is possible then, that the heavy departing aircraft need not be the last one to take off in order to achieve maximum throughput on the departure runway. For example, assume that the arrival sequence for four (4) landings cannot be altered and is (as supplied by TRACON for example):

S (210, 0.5) - S (270, 0.5) - L (330, 1) - S (390, 0.5)

where the two numbers in parentheses are explained as follows:

- a) The first numbers are the scheduled touchdown times (landing separations are assumed to be 60 sec between successive arrivals) and
- b) The second numbers are the units of taxiway space that each aircraft is expected to occupy depending on its weight class (0.5 for small, 1 for large and 1.5 for heavy aircraft).

In this example, we assume that only crossing point X2 can be used with a total taxiway capacity of two (2) units and therefore, all arriving aircraft have to gather at X2 awaiting crossing clearance. If no arrival schedule modifications are allowed, before all taxiway capacity is used,

X2 will accommodate arrivals in the sequence:

$$S (210, 0.5) - S (270, 0.5) - L (330, 1)$$

At this point, the capacity of X2 is saturated and all three aircraft must be cleared to cross in order to allow for the following small aircraft S (390, 0.5) to land and occupy taxiway space at X2 after clearing the arrival runway.

Assuming that the last aircraft to land (before crossings are cleared) will occupy the runway for 50 sec, all three crossings will be available to cross at time point $330 + 50 = 380$. Consistent with heuristic rules solicited through interviews with experienced air traffic controllers, it is assumed that the time necessary for crossings to be completed is 40 sec for the first and 10 sec for each aircraft following. Therefore, the three crossing aircraft will cross the departure runway in $40 + 10 + 10 = 60$ sec.

Consequently, if the departure sequence is left unchanged (S - S - S - L - L - H), at time point 380, the first large aircraft of this sequence will take off occupying the departure runway for an (assumed) period of 50 sec and then the same runway will need to be blocked for an additional 60 sec for the crossings. Therefore, the next large aircraft of the departure sequence will be allowed to take off $50 + 60 = 110$ sec later at time point $380 + 110 = 490$ and the final departure sequence will be (Case 1):

$$S (200) - S (260) - S (320) - L (380) - \mathbf{X} - L (490) - H (550) - N (550 + 90 \text{ or } 120 = \mathbf{640 \text{ or } 670})$$

where X represents the time point when crossings are cleared and N is the next aircraft taking off

right after the last aircraft of this departure group, which will be separated from the last heavy (H) aircraft by 90 sec if it is also a heavy or by 120 sec if it is a large or a small departure.

Given that the wake vortex separation behind the large aircraft is 60 sec, there is an additional $110 - 60 = 50$ sec that is added to the departure schedule due to the crossings. Also, the heavy departure still remains to be served (take off) incurring at least 90 sec of wake vortex separation to the departure runway.

On the other hand, assume that the original First Come First Serve departure schedule of the first six aircraft is changed from

S - S - **H** - S - L - L

to

S - S - **S** - **H** - L - L

as opposed to

S - S - S - L - L - **H**

In this case, at time point 380 the heavy aircraft is cleared to take off ahead of the two large aircraft. Under the same assumptions of 50 sec runway occupancy times and 60 sec for three crossings to be completed, the next time a departure will be allowed to take off is again at time point 490. However, 120 sec of wake vortex separation have to be allowed between the heavy and the following large. The advantage in this case is that the heavy aircraft has already been serviced at this point and the departure schedule is (Case 2):

S (200) - S (260) - S (320) - H (380) - **X** - L (500) - L (560) - N (**620**)

The following aircraft N is then able to take off only 60 sec after the last large in order to maintain the required wake vortex separation. Therefore, in Case 2, after crossings are considered, there is still at least $640 - 620 = 20$ sec of runway-time savings compared to Case 1 (higher departure throughput). This happens because the heavy aircraft of this departure group was not left last to take off and 110 sec out of the 120 sec of wake vortex separation time behind it were used for crossings.

So far, in the above examples, it was assumed that the arrival schedule cannot be altered and arrivals were simply treated as additional (but fixed) requests for runway time. However, changes in the arrival schedule are possible. In fact, the example below will demonstrate that weight class sequence planning can offer benefits to crossing aircraft that may have to absorb less delay at crossing points, if modifications to the landing sequence and timing are allowed.

Arriving flights reach the runway in a certain sequence that is predetermined by TRACON controllers. At some airports this sequence is determined by advanced Air Traffic Management technologies (e.g. CTAS). If no changes are permitted, arriving aircraft inevitably limit the amount of runway time that is left to be allocated to departures and crossings. On the other hand, allowing changes in the arrival schedule may provide flexibility in producing ROP solutions with higher departure and arrival throughput and thereby enable solutions that are closer to “global” optimality. A flexible arrival schedule may also be particularly useful in the event that the

heuristic algorithm described above cannot reach a feasible solution. In such a case, adjusting the arrival schedule may help the algorithm to produce a feasible runway operations schedule.

Even though it is beyond the scope of this thesis, it must be noted that it might be possible to design the optimization heuristics so that the algorithm alters the arrival schedule before solution infeasibility is reached. This can happen by incorporating information about the arrival stream into the optimization algorithms and by taking into account the types of aircraft expected to arrive and request crossing time on the departure runway, the airport geometry and the taxiway capacity constraints. Consequently, crossing operations can become “smarter” and the runway schedule results can be closer to runway throughput optimality. The following example demonstrates one of the possible ways in which the arrival schedule can be linked to crossing and departure operations.

In the example airport system of Figure 5.1, there is limited taxiway space for holding aircraft on two connecting taxiway segments between the two runways (X1 and X2 in Figure 5.1). The maximum number of aircraft allowed between the runways is predetermined (this can be a simulation test parameter) for each crossing point and depends on the weight class of the aircraft present. Initially, we assume that all small (S) aircraft can exit the runway early enough to make it to cross-point X1 and that all other aircraft (large and heavies) use the other point X2. In some instances, such an assumption may be relaxed in the interest of “smart crossings.”

Based on the arrival aircraft classes in hand and the crossing point capacities, the problem is to

design an arrival sequence that brings arrivals to the crossing points in such a way that no cross-point capacity is wasted due to saturation of another crossing point. For example, assume that:

- Small aircraft occupy one half (0.5) unit capacity, large occupy one (1) and heavies occupy one and a half (1.5) units,
- Both cross-points have a capacity of two (2) units and
- The arrival sequence for five (5) landings (as supplied by TRACON) is:

S (210, 0.5) - S (270, 0.5) - L (330, 1) - S (390, 0.5) - H (450, 1.5)

where the numbers in parenthesis are the scheduled touchdown times (one minute apart from each other) and the taxiway capacity that each aircraft is expected to occupy. If no arrival schedule modifications are allowed (Case 1), under the assumption made earlier, cross-point X1 will accommodate all small aircraft in the sequence:

S (210, 0.5) - S (270, 0.5) - S (390, 0.5)

with a utilized taxiway capacity of one and a half (1.5) units out of a total of two (2). Cross-point X2 will then have to accommodate one large and one heavy aircraft in the sequence:

L (330, 1) - H (450, 1.5)

However, when the large aircraft lands, there will be no taxiway space left for the heavy ($1+1.5 > 2$). Positioning the large ahead of the heavy saturates point X2 earlier and therefore one half (0.5) capacity unit in X1 ($1.5 < 2$) and one (1) capacity unit in X2 ($1 < 2$) are wasted for this group of arrivals.

Linking the Departure Planner decision-aiding tool and the resulting departure schedules to the

arrival stream can improve runway and taxiway space utilization. If, for example (Case 2), the arrival sequence is altered from:

S (210, 0.5) - S (270, 0.5) - **L (330, 1)** - S (390, 0.5) - **H (450, 1.5)**

to:

S (210, 0.5) - S (270, 0.5) - **H (330, 1.5)** - S (390, 0.5) - **L (450, 1)**

and we also assume that small aircraft can continue their landing roll all the way to cross-point X2 if necessary and also large aircraft can exit early to cross-point X1 if commanded to do so, then the two cross-points can receive aircraft in the following order:

Cross-point 1: S (210, 0.5) - S (270, 0.5) - L (450, 1)

Cross-point 2: H (330, 1.5) - S (390, 0.5)

without wasting taxiway capacity at all. The “swapping” between the large and the heavy aircraft of the group allows for all arriving aircraft to be accommodated in the taxiway space between the two parallel runways and in that way all five aircraft can be crossed at the same time, with the total crossing time being exactly the same as in Case 1 and under the same crossing clearance.

The examples presented above illustrated the departure runway throughput benefits that may be achieved when crossing aircraft are included in the planning process. In addition, the two planning examples that include arrivals (with or without changes to the arrival schedule) illustrate the advantage of solving the broader planning problem that includes all kinds of operations on the same runway.

5.6.2 Stage Two

As stated previously, the goal of the 2nd stage is to assign specific aircraft to the weight class slots defined in the 1st stage. Assuming that the Target Class Sequence from the 1st stage is:

$$S - S - H - s/s/l - S - L - L - L - s - L - S$$

then the goal is to assign each of the nine (9) departing aircraft under consideration to a class slot. If the earliest time that any small aircraft from the departure group can be at the runway is 670, then, based on wake vortex separations between class slots and on landing and crossing runway occupancies, the following set of times corresponds to the start times of the class slots in the Target Class Sequence:

$$670 - 730 - 790 - (\mathbf{X}) - 910 - 970 - 1030 - 1090 - (\mathbf{X}) - 1190 - 1250$$

These time values can be used to calculate the time to complete all departures and therefore the final departure throughput as the inverse of that time. We assume the time that a given aircraft takes off to be equal to the midpoint of the slot to which it is assigned. For the slots in the Target Class Sequence, these times are:

$$700 - 760 - 820 - (\mathbf{X}) - 940 - 1000 - 1060 - 1120 - (\mathbf{X}) - 1220 - 1280$$

Also, the earliest time that the same aircraft could have taken off (if there were no departure queues) is equal to the time it is ready for pushback plus its unimpeded taxi time. For the calculation of the values of the delay-based objective function, the delay for each aircraft is equal to the difference between the midpoint of the assigned slot and the earliest time that aircraft could have taken off. Indicative of the complexity of the 2nd stage optimization problem is the fact that there can be many feasible ways to assign the nine scheduled departing flights to the

nine class slots of the Target Class Sequence and in fact some of them may lead to the same final runway throughput and the same cumulative departure delay. However, not all feasible solutions will necessarily be of the same quality in terms of other problem constraints, such as fairness for example. Two example solutions from the 2nd stage of the optimization algorithm for this small assignment problem of nine (9) aircraft are given in Table 5.1.

Flight Number	Weight Class	Pushback Pos.	Case 1		Case 2	
			Take off (MPS = 3)	Max. Delay (min)	Take off (MPS = 1)	Max. Delay (min)
USC168	S	3	1	1	2	2
COA339	S	1	2	1	1	0
AAL1317	H	2	3	2	3	2
<i>Crossings</i>	<i>s/s/l</i>					
N109FX	S	5	4	3	4	3
DAL1821	L	6	5	3	6	4
SGR501	L	7	6	3	7	4
USA1854	L	4	7	6	5	4
<i>Crossings</i>	<i>s</i>					
USS6171	L	9	8	5	8	5
N180M	S	8	9	6	9	6
<i>Total</i>				<i>30</i>		<i>30</i>

Table 5.1: Example “aircraft to class slot” assignments

Both solutions are feasible “aircraft to class slot” assignments that the second stage optimization generates but they are based on two slightly different sets of constraints. In both cases, the same ATC operational constraints are satisfied in addition to all other physical and “slot sequence” constraints that are present in the problem. However, Case 1 is less strict in terms of fairness (MPS = 3) than Case 2 (MPS = 1) and that is why shifts of up to 3 positions between pushback (column 3 in Table 5.1) and takeoff (column 4 in Table 5.1) are part of the feasible solution. The difference in position shifts between the column pairs “3 & 4” and “3 & 6” evidently show that

Case 2 is a more fair allocation of aircraft to slots, even though the two cases have all else equal, i.e. the final throughput and cumulative departure delay remain the same.

Chapter 6

SIMULATION TESTS & RESULTS

The “two-stage” heuristic optimization algorithm described in Chapter 5 was implemented using the Matlab programming language and Simulink simulation tools. The simulation test bed described in Chapter 5 was then used to study the behavior of each of both stages of the algorithm and determine the operational benefits that would be gained by using the algorithm for runway operations planning. The robustness of the algorithm (i.e. change in benefits in response to changes in aircraft pushback times) and the stability of its output (i.e. changes in aircraft and weight class sequences in response to changes in aircraft pushback times) were also studied and the results of these studies are presented in sections 6.4 and 6.5. Ideas for future research work to enhance the model will be given in Chapter 7.

6.1 Demand Traffic Scenarios

The arrival and departure schedule that was used as the initial input to the simulation model

was derived from traffic data collected during observations from the air traffic control tower at Boston Logan airport on February 2, 1999. The 4-hour data set was collected during the busy morning hours between 6 am and 10 am, when there was high pressure on the departure runway, because such busy periods are typically when air traffic controllers can benefit the most from a decision-aiding tool for surface traffic management.

Due to limitations in physical space in the tower and access to voice communication channels used by air traffic controllers, the data set was not complete. Specifically, the requisite flight information and time stamps of key events associated with departure and arrival operations were only obtained between 6 am and 7 am. Fortunately, the schedule of arrivals during this hour that interacted with the departure flow on the airport surface had no “gaps” and therefore, there was also high pressure on the arrival runways. This ensured constant pressure of crossing requests and minimized the possibility of runway “idle” time in the final schedule of operations (departures & crossings) on the departure runway.

All simulation tests were conducted using the benchmark airport and runway configuration presented in Figure 5.1. This was a good match to the data set because all the data came that was collected from a time period when the airport was operating under the runway configuration 22L / 22R / 27²⁷, which is very similar to the one represented in Figure 5.1. It is important to note that, the departure and arrival schedule derived from the data set collected at Logan was only used to analyze the potential operational benefits that would be gained by using the two-stage algorithm (sections 6.3.3 and 6.3.4). In addition to the Logan schedule, other input schedules

²⁷ More on Boston Logan runway configurations can be found in [74]

with specific characteristics were also created and used depending on the nature of each particular study, as it will be explained in the corresponding sections (6.4, 6.5, 6.6 and 6.7).

Regarding the simulation output, a description of its format for each of the two stages of the runway operations planning algorithm can be found in Appendix D. There, details are given on how the simulation output for each stage is generated, as well as how runway crossings affect the output of the 1st stage.

6.2 Evaluation of Algorithm Performance

The behavior of the simulation output, namely the optimized runway operations plans, was examined under various airport operational schemes in order to evaluate the overall utility of the two-stage planning algorithm. More specifically, the following were determined:

1. *Benefits*: The improvement in operational performance metrics, such as departure throughput and cumulative²⁸ queuing delays that would be achieved when the two-stage planning algorithm is applied,
2. *Robustness*: How the airport throughput and delay performance achieved through planning change in the presence of varying levels of uncertainty in the gate pushback schedules that are given as an input to the planning algorithm and
3. *Stability*: How the suggested weight class sequence and aircraft takeoff sequence change in

²⁸ i.e. over all aircraft involved in the planning ...

the presence of input schedule uncertainty.

6.3 Benefits Analysis: A Representative Example Case - Boston Logan Airport

A research question that usually arises when a decision-aiding tool is being designed is: “*Can such a decision-aiding tool provide its users with operational suggestions that are beneficial (with respect to pre-determined performance metrics) when compared to operational scenarios with no decision-aiding available?*” The following example of a representative traffic situation at Boston’s Logan airport is presented in an effort to answer this research question. Through this example, we outline the methodology that was developed to determine the potential airport operational benefits of deploying the two-stage algorithm in a ground operations decision-aiding tool. Benefits analysis is presented under two different situations: one that has no air traffic flow restrictions affecting airport operations and a second one with active flow restrictions.

Furthermore, this methodology will be used in later sections with various departure demand profiles as in input, in order to investigate the extensibility of the two-stage planning algorithm to varying traffic situations.

The main objective of introducing the two-stage runway operations planning algorithm is to provide decision support to air traffic controllers who are handling departing traffic as it interacts

with the arriving traffic on the airport surface. Hence, the following were examined:

1. How the plans suggested by the two-stage algorithm perform when compared to runway sequences generated by air traffic controllers without any planning.
2. If and how the above performance comparison (Planned vs. FCFS) is affected by the presence of various types of system constraints, such as air traffic flow restrictions (e.g. Miles In Trail, EDCTs).

An example of a case where no planning is performed is the servicing of aircraft at the gates and at the runway on a First Come (Call Ready for Pushback or Arrive at Runway event) First Serve (Clear for Pushback or Release to Take Off event) basis. In real-world operations, air traffic controllers do not entirely limit themselves to “naïve” aircraft handling strategies such as First Come First Serve (FCFS). However, the latter is, in most cases, realistic since it is identical or very similar to the traffic handling strategies that controllers deploy during high traffic periods at busy US airports. Also, the FCFS regime offers a simple strategy that is unbiased relative to any specific aircraft sequencing and control strategies that controllers may use at particular airports and can therefore offer a valid comparison basis for all simulation tests performed.

A separate simulation of departure operations under a First Come First Serve (FCFS) aircraft handling strategy was created to compare the throughput and delay performance when there is: i) advance planning (using the two-stage algorithm) and ii) no planning. This simulation is referred to as the “FCFS simulation” for the remainder of this document.

In this FCFS simulation there is no re-ordering of the takeoff sequence and no change of departure times to meet a plan. Rather, a First Come First Serve scheduling logic is followed, which schedules departing aircraft to take off in the sequence they arrive with at the runway. The takeoff times are set as mandated by the traffic that each departing aircraft encounters ahead in the runway queue and by the arriving aircraft that land on a parallel runway and become crossing requests on the departure runway²⁹. Only wake vortex separations and air traffic control restrictions (when applicable) are obeyed, but no throughput maximization or minimization of time-based performance metrics is attempted.

6.3.1 Airport Operational Performance Metrics

Performance is evaluated in terms of operational metrics that make intuitive sense to air traffic controllers, such as departure and total throughput, or the following two time-based performance metrics used are:

- *Queuing Delay of an aircraft*: defined and calculated as the difference between the aircraft's actual takeoff time and the earliest time that the aircraft could reach the runway, which is also the earliest time it could take off if there was no runway queuing and
- *Elapsed Time of an aircraft*: defined as the time that elapses from the moment an aircraft pushes back from its gate until the aircraft is cleared to take off. Part of this elapsed time is the taxi out process of the aircraft and the remaining is the time the aircraft spends at the ramp area (if it exists) and at the runway takeoff queue.

²⁹ Remember that, arriving traffic was assumed to have no "gaps".

6.3.2 Methodology for Evaluating Planning Benefits

In this example, the schedule of pushback times that was entered as an input to the simulation is a subset of 15 aircraft extracted from the Boston Logan data set that was mentioned earlier. It has a fleet mix of 13.33 % Heavies (2 aircraft), 66.67 % Large (10 aircraft) and 20 % Smalls (3 aircraft). Note that this is not a set of pushback times that are exactly as originally published. Even though there was no gate or runway operations planning performed at Logan airport when this data was collected, due to pushback uncertainties, some of the times in the data set ended up being different than the ones originally published and distributed to the tower controllers ahead of time. The originally published pushback times could be extracted from the Airline Service Quality Performance (ASQP) database but only for the jet operations that are recorded there. Therefore, since there were also non-jet operations in the morning when the Logan data was collected, we chose to use the set of pushback times that actually occurred as an input to the simulation. That schedule is referred to as the “Original Pushback Schedule” at the top left of Figure 6.1, which is a schematic representation of the “Planner” as it will be referred to in the remainder of this chapter, namely the planning framework that uses the Runway Operations Planning algorithm. The framework will be explained in full in the following paragraphs.

The simulation was run once for the specific original pushback schedule input. First, in order to bring all involved aircraft to the runway, that original schedule was propagated forward based on average unimpeded taxi times, which depend on the weight class of each aircraft and the terminal it is coming from. This “forward propagation” created an “expected aircraft sequence and timing at the runway” which was then used by the two-stage ROP algorithm to generate a runway operations plan, namely the Planned Takeoff Schedule at the top right corner of Figure 6.1.

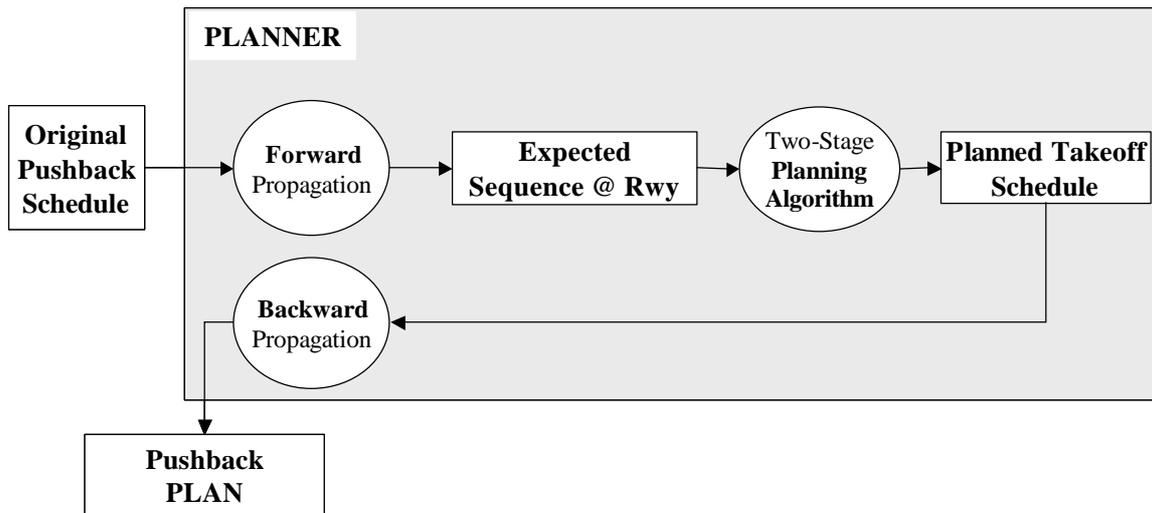


Figure 6.1: The “Planner” - Planning Gate Pushback Operations by Using the Results of the Runway Operations Planning Algorithm

This Planned Takeoff Schedule had to somehow be conveyed to the air traffic controllers in the airport tower in a useful form. Since it is always easier for controllers to understand and use planning suggestions in terms of gate pushback operations as opposed to suggestions that come in the form of takeoff sequences and times, the Planned Takeoff Schedule was finally propagated backwards to the gates based on the propagation methodology outlined in Chapter 4. The outcome of this back-propagation is the Pushback Plan at the lower left corner of Figure 6.1.

An important part of the back propagation methodology from Chapter 4 is to determine the coefficients of the quadratic queuing curve to be used in “translating” the runway operations plan (Planned Takeoff Schedule) to a gate operations plan (Planned Pushback Schedule). Starting from the original queuing curve that was elicited from the queuing data in [74], different values for the curve coefficients were tested until a curve was found that yielded lower (compared to the original curve) average values for the time-based performance metrics (cumulative queuing

delays and elapsed times) without compromising the achieved departure throughput values. This is necessary in order to determine the curve that generates the appropriate back-propagation queuing adjustment which plans each aircraft to push back early enough to ensure a sufficient level departure pressure on the runway, while at the same time it does not allow the resulting queuing delays and elapsed times to increase uncontrollably.

The starting point of the back-propagation is always the original Planned Takeoff Schedule generated by the ROP algorithm. However, each time a different set of curve coefficients is tested in the back propagation process, the Planned Takeoff Schedule is propagated back to the gates as a different Pushback Plan, which is then submitted as a planning suggestion to the air traffic control tower. It is then assumed that the controllers use that pushback plan as a planning guide and handle the taxiing process with no additional planning suggestions available. Therefore, in order to estimate the throughput and delay performance achieved through each queuing curve and the corresponding pushback plan that is provided to the controllers, the Planned Pushback Schedule from Figure 6.1 has to be used as an input to the FCFS simulation that was mentioned earlier. That was actually done in this case and a representation of the entire testing framework, which includes the Planner from Figure 6.1 and how its output is used as an input to the FCFS simulation, is shown in Figure 6.2.

The FCFS simulation was run in a Monte Carlo setting with several thousands of iterations, where each such iteration corresponds to an instance of a set of randomly sampled taxi times (Taxi Randomness in Figure 6.2) for the set of aircraft involved. The results from all Monte Carlo iterations are depicted as the overlapping perforated frames at the lower right corner of

Figure 6.2. Each set of Monte Carlo runs corresponds to a different queuing curve being evaluated (different Backward Propagation in the Planner of Figure 6.1). After completing each set of Monte Carlo runs, statistical analysis of the departure and total throughput as well as the departure delay and elapsed time metrics over each entire set was performed. The statistical results are shown as “Mean1” and “StDev1” at the lower right corner of Figure 6.2.

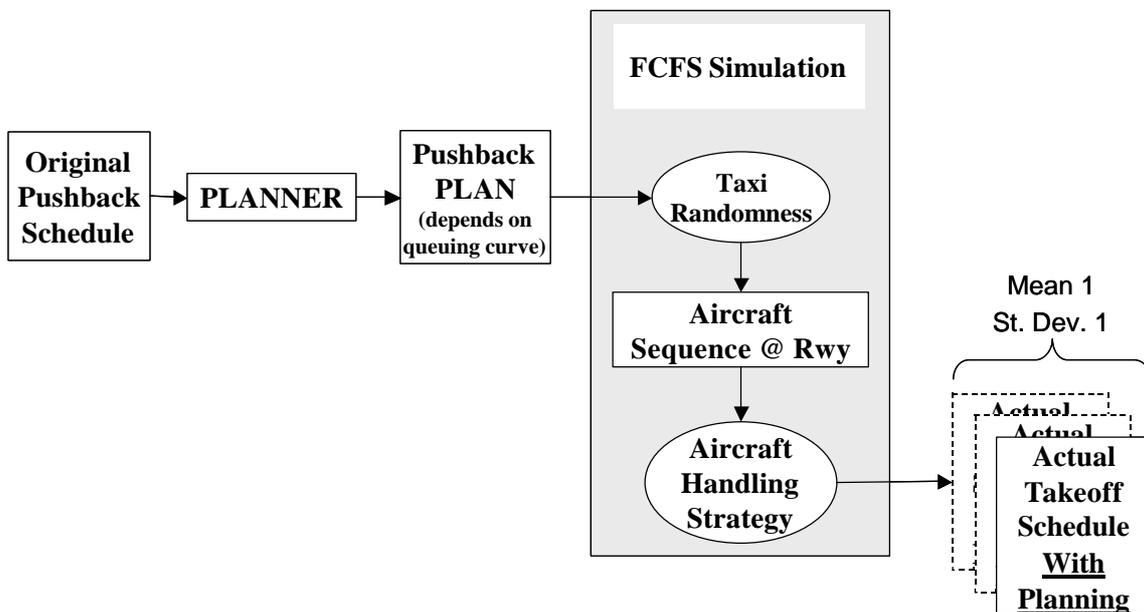


Figure 6.2: Using the “Planner” and the FCFS simulation for evaluating different queuing curves for the back-propagation queuing adjustment of the Planner (Figure 6.1)

Using the particular pushback schedule extracted from Logan data as the “original pushback schedule” input to the procedure in Figure 6.2, at first, an optimal runway operations plan was generated which, if followed exactly, is expected to yield a departure throughput of 45 departures / hour and a total throughput (including crossing operations on the departure runway) of 93 operations per hour. This runway operations plan was then back-propagated to the gates by initially using the “Original queuing curve” resulting from the Logan data (Idris’s work [74]) that

was first presented in Figure 4.4. A set of Monte Carlo runs was then completed and average values for performance metrics, such as departure throughput and cumulative taxi times were obtained. However, if the original queuing curve is used for all testing purposes an implied assumption is made that the queuing behavior that manifested at Logan airport on the day that the queuing data in Figure 4.4 was collected is applicable on any other case. Since there was no research evidence to make such an assumption a valid one, the only aspect of this queuing behavior that was kept for all tests was the quadratic trend. Beyond that, the same process (Figure 6.2) was repeated and other quadratic queuing curves with different coefficients were evaluated.

Each new curve that was evaluated was generated using the following method:

- a) The original set of data points was used as a seed but only the first and last 2 to 3 data points were used this time.
- b) The first few points were used with the exact same y-values, because these are the “throughput drivers” as it was explained in section 4.2 and the objective for each new curve generated was to not compromise throughput compared to the original curve.
- c) On the other hand, the y-values of the last few data points from the original set of data were not kept the same but instead they were used only as a basis to generate new data points with lower y-values³⁰, which corresponded to lower queuing time values. This was done with the objective of reducing the total amount of queuing delay that the set of aircraft absorbs by reducing the amount of queuing time that we expect the last few aircraft to absorb.

³⁰ Reduction of the original y-values was by a specific pre-determined amount each time a new curve was generated.

After comparing the statistical results for all curves, one with lower delays (cumulative for all 15 involved aircraft) and approximately the same throughput performance as the original queuing curve was selected to be used in the “planned takeoff time to planned pushback time” back-propagation. The selected curve is shown in Figure 6.3 together with the original queuing behavior data points from Idris’s work [74]. The curve in Figure 6.3 was selected because it has the following features:

- a) Average throughput performance (at the end of the testing process in Figure 6.2) is not compromised when switching from the original to the new queuing curve (as shown in Table 6.1), because the queuing behavior of the first few aircraft, which are the main throughput “drivers” as explained earlier, is almost identical between the two curves and
- b) Cumulative delays (runway queuing times) and cumulative elapsed times are reduced by about 12 minutes in total for all 15 aircraft involved (shaded values in Table 6.1 correspond to the final curve).

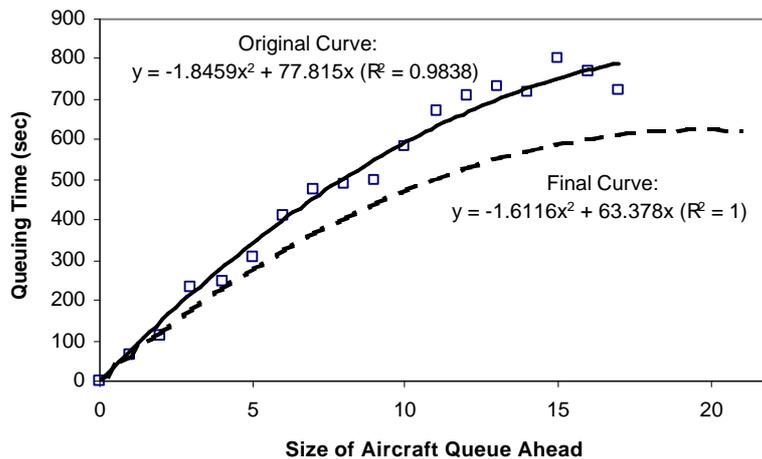


Figure 6.3: Final vs. Original Queuing Curve

	Original Queuing Curve	Final Queuing Curve
Dep. Runway Throughput (departures / hour)	40.1	39.92
Total Runway Throughput (operations / hour)	90.05	89.75
Cumulative Dep. Delay (min)	61.15	48.89
Cumulative Elapsed Time (min)	249.69	

Table 6.1: Performance Comparison between Using the Original and the Final Queuing Curve in Planning

At this point, having determined the quadratic queuing curve to be used in the “Backward Propagation Assumptions” block of Figure 6.2, the pushback planning suggestions (Pushback Plan in Figure 6.1 and Figure 6.2) are to be used for the benefits analysis. The latter is based on comparing the airport operational performance under the following two scenarios:

1. Aircraft push back according to their original pushback times and
2. Aircraft push back according to a pushback plan.

It is assumed that in both scenarios air traffic controllers handle departures on a First Come First Serve basis after they push back from their gates and enter the taxiway system. In order to simulate the above two scenarios, the FCFS simulation was provided with the following two different schedule inputs, as shown in Figure 6.4:

- a) The original pushback schedule (“Without Planning” case in Figure 6.4) and
- b) The Pushback Plan that was generated using as a guide the runway operations plan suggested by the ROP algorithm (“With Planning” case in Figure 6.4, which was also depicted in Figure 6.2 but with more details about the FCFS simulation).

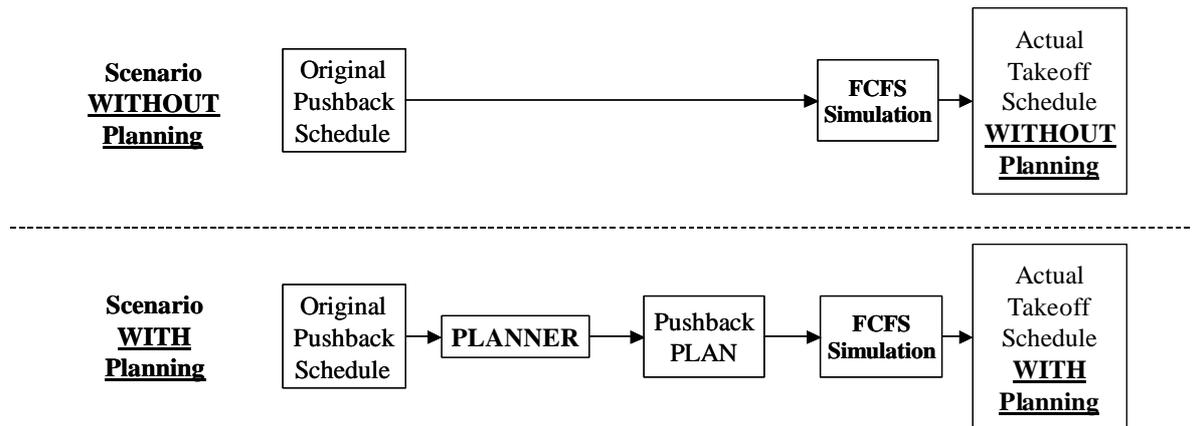


Figure 6.4: Comparing Airport Operational Performance between Scenarios WITH and WITHOUT Planning

Similar to the ROP simulation, the FCFS simulation propagates (“translates”) its input (planned or original gate pushback schedule) forward through the airport taxiway system to an aircraft sequence at the runway. Unlike the forward propagation in the planner (Figure 6.1), this forward propagation, as part of the FCFS simulation that attempts to approximate real world ground operations, takes into account the inherent uncertainty that exists in airport taxi operations (Taxi Randomness in Figure 6.2). Real-world taxi uncertainty was simulated by randomly sampling for each involved aircraft a taxi time (from gate pushback until entering the runway queue) from a set of pre-determined taxi time stochastic distributions. These distributions were derived from a statistical analysis of the taxi time data that was collected at Boston’s Logan air traffic control tower on February 2, 1999, by monitoring for several hours under runway configuration 22L / 22R / 27 the communication events between controllers and pilots and recording their times [74].

The entire benefits analysis methodology is presented in Figure 6.5. For each of the two different inputs in Figure 6.4, the FCFS simulation was run in a Monte Carlo setting with several

thousands of iterations in each case, just as it was done when different queuing curves were being evaluated for the back propagation process. As in the process of Figure 6.2, the Monte Carlo runs are depicted as the overlapping perforated frames at the right side of Figure 6.5. Each such run corresponds to an instance of a set of randomly sampled taxi times for the set of aircraft involved. For each of the two different cases in Figure 6.4, statistical analysis of the departure and total throughput as well as the cumulative departure delay and elapsed time metrics over each entire set of Monte Carlo iterations is performed and the statistical results (presented as “Mean1”, “Mean2” and “StDev1”, “StDev2” at the lower right corner of Figure 6.5) are then compared.

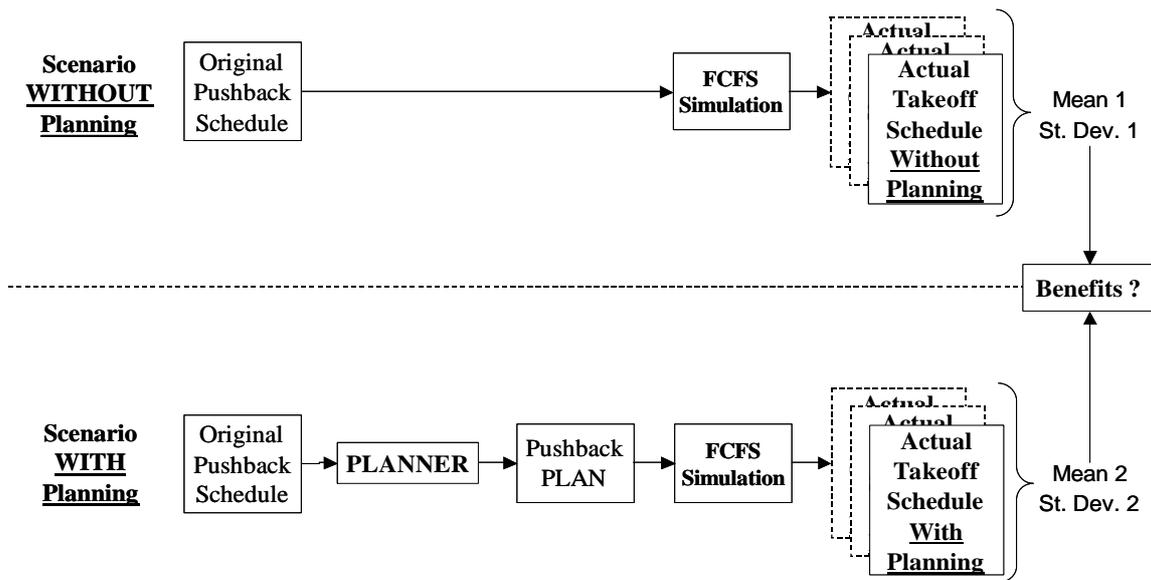


Figure 6.5: Overall Benefits Analysis Methodology - Comparing Airport Operational Performance between Scenarios WITH and WITHOUT Planning under a FCFS Ground Traffic Handling Strategy

If we recall the conceptual architecture for the suite of tools that should be included in a departure planning decision-aiding system, which was introduced in [1] and described in

Appendix B, we can argue that the above testing methodology is based on a simplified version of that conceptual architecture, with fewer subcomponents as this is presented in Figure 6.6.

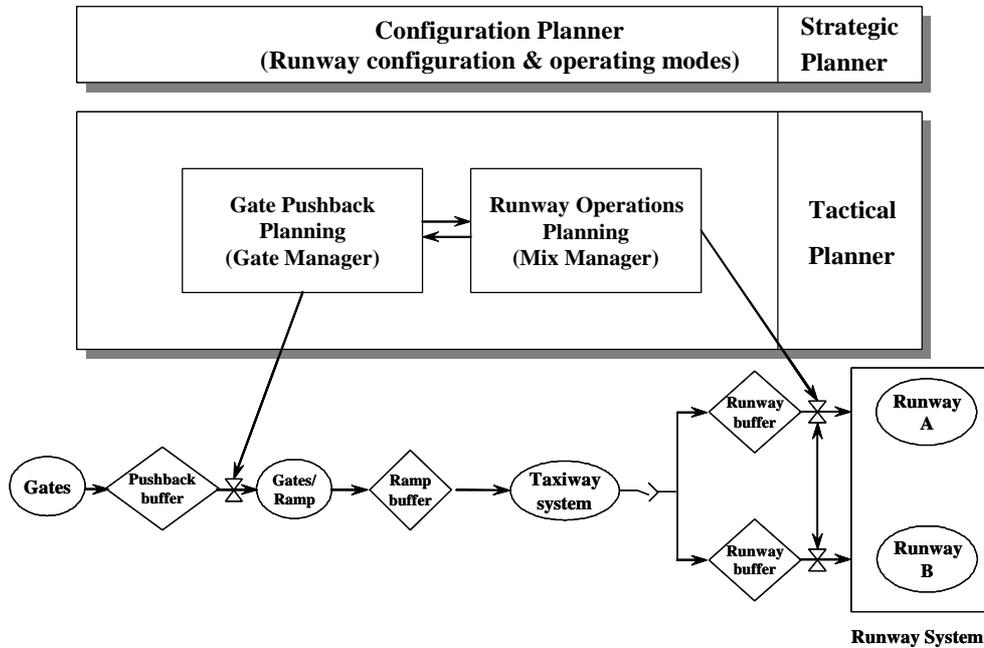


Figure 6.6: Conceptual Architecture of the Departure Planner – Simplified Version

Next, using the selected back-propagation queuing curve in Figure 6.3, results of the benefits analysis under different assumptions regarding air traffic control restrictions are presented.

6.3.3 Potential Benefits from Planning in the Absence of Air Traffic Flow Restrictions

At first, benefits analysis, as described in Figure 6.5, was performed for a case when no air traffic control restrictions or other flow constraints are active. The effect that air traffic control restrictions may have on operational benefits from planning runway operations will be examined

in the next section. A particular example is presented with Miles In Trail spacing being the active flow restriction, but the insight gained is applicable to cases with other types of flow restrictions are active, such as Expected Departure Clearance Time (EDCT) or Departure Sequencing Program (DSP) time windows.

Starting from the same original pushback schedule of 15 aircraft that was mentioned in section 6.3.2 (15 aircraft with a fleet mix of 2 Heavies, 10 Large and 3 Smalls) and following the procedure of Figure 6.5, the FCFS simulation was run for 5,000 Monte Carlo iterations for each of two different inputs: i) the pushback plan that was created by the Planner and ii) the original pushback schedule. In each case, the 5,000 iterations created a data set large enough to provide statistically significant results, which are shown in Figure 6.7 and Figure 6.8.

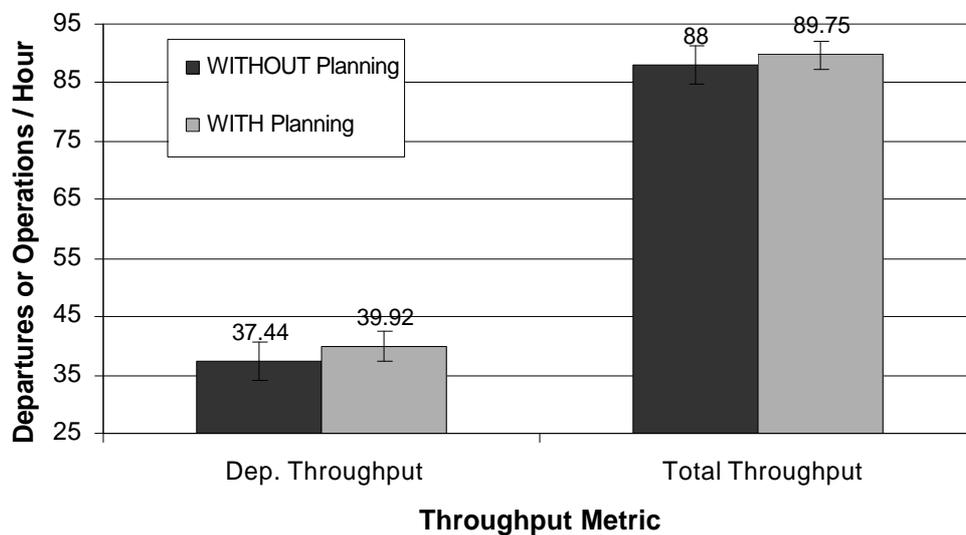


Figure 6.7: Departure and Total Throughput

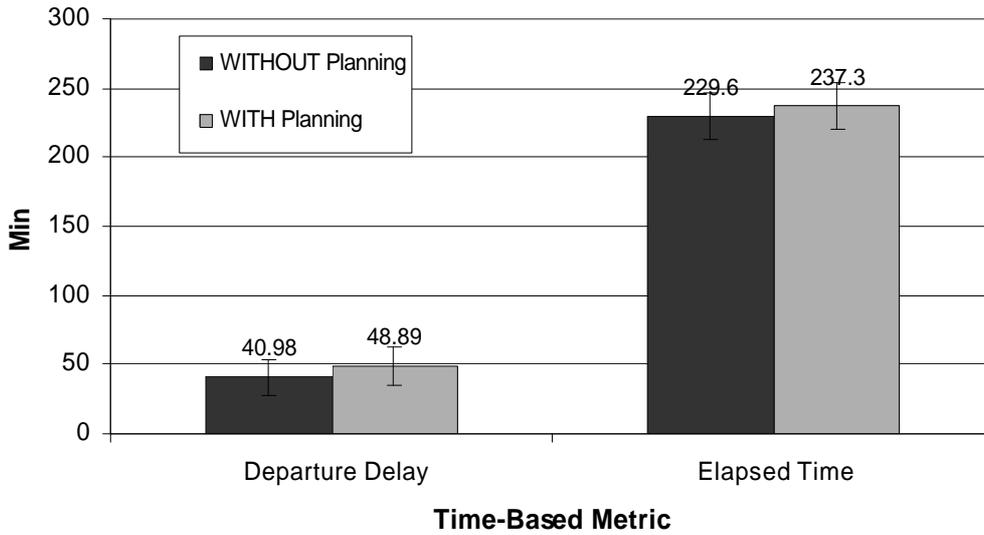


Figure 6.8: Cumulative Departure Delay and Elapsed Time

The above results are also summarized in Table 6.2.

	Original Pushback Input	Planned Pushback Input (Final Queuing Curve)	BENEFITS	
			YES	NO
Dep. Runway Throughput (departures / hour)	37.44	39.92	≈ 2.5 departures/hour	
Total Runway Throughput (operations / hour)	88	89.75	1.75 operations/hour	
Cumulative Dep. Delay (min)	40.98	48.89		≈ 30 sec / aircraft
Cumulative Elapsed Time (min)	229.6	237.3		≈ 30 sec / aircraft

Table 6.2: Benefits Analysis for Traffic Scenarios WITHOUT Flow Restrictions (Earlier than Scheduled Pushback Times ARE Allowed)

In the case without planning, where the original pushback schedule was used as an input to the FCFS simulation, average departure throughput (over 5,000 Monte Carlo runs) was 37.44 departures / hour and average total throughput was 88 operations / hour. When runway

operations planning was performed and the optimal runway operations plan was used as a guide for pushback operations³¹, there was a gain of about 2.5 departures / hour (= 39.92 - 37.44). However, in this particular example, the throughput gain achieved through runway operations planning comes to the expense of small additional delays absorbed by each aircraft. Recall that the two-stage planning algorithm was constructed based on the selection of “demand weight class mix” as the decomposition pivot element and therefore tends to prioritize on improving the departure throughput performance of the system. Consequently, given the tradeoff between runway throughput and delays that was discussed in Chapter 4, for this particular example, the planned pushback schedule input yields slightly higher cumulative delays and elapsed times. In fact, given that 15 departures are involved in the planning, the extra delay that each aircraft has to absorb on average when runway operations planning is performed is about $(48.89-40.98)/15 = 0.5$ min per departing aircraft and the additional elapsed time is also about $(237.3-229.6)/15 = 0.5$ min per departing aircraft.

In the above results, when the pushback plan was generated by back-propagating the runway operations plan, 6 of the 15 aircraft were required to push back slightly earlier than their original pushback time and only one of them was required to push back about 5 minutes earlier. Over all aircraft that were planned to push back earlier the average difference between their planned and original pushback times was only 1.62 minutes. It turns out that, NOT allowing planned pushback operations to occur earlier than original pushback times, defeats the purpose of actually planning pushback times in the first place, because the previously achieved throughput benefits are decreased. On the other hand, if aircraft are not planned to push back earlier, understandably

³¹ by translating it to a gate operations plan using the final queuing curve from Figure 6.3 ...

the total elapsed time and the total time the aircraft spend waiting at the runway queue are decreased. This reverses the previous results and introduces small delay and elapsed time benefits from planning but compromises some of the throughput benefits previously reported (Table 6.3).

	Original Pushback Input	Planned Pushback Input (Final Queuing Curve)	BENEFITS	
			YES	NO
Dep. Runway Throughput (departures / hour)	37.44	38.4	≈ 1 departure / hour	
Total Runway Throughput (operations / hour)	88	88.65	≈ 0.65 operations / hour	
Cumulative Dep. Delay (min)	40.98	37.29	≈ 15 sec / aircraft	
Cumulative Elapsed Time (min)	229.6	225.9	≈ 15 sec / aircraft	

Table 6.3: Benefits Analysis for Traffic Scenarios WITHOUT Flow Restrictions (Earlier than Scheduled Pushback Times ARE NOT Allowed)

The remaining test results presented here were generated under the assumption that earlier than originally scheduled pushback operations ARE allowed.

6.3.4 The Effect of Air Traffic Flow Restrictions on Potential Planning Benefits

When air traffic flow restrictions are introduced in the Runway Operations Planning problem, the resulting runway operations plan deviates from a FCFS takeoff sequence or even from the optimal takeoff sequence that was derived for the unrestricted case, because usually, takeoff sequence positions of specific aircraft need to change for the flow restrictions to be satisfied. The difference in runway operations plans under a restricted vs. an unrestricted traffic scenario is

demonstrated in Table 6.4 for the same Logan example of 15 aircraft. The objective function used in both scenarios was the same mixed “elapsed time & fairness”-based objective function (equation (D.1) in Appendix D).

Scheduled Pushback Position	Take off Pos. Plan No Active Flow Restrictions	Take off Pos. Plan Miles In Trail: EWR: 35 MIT, JFK: 30 MIT	Destination of Aircraft Affected by Miles In Trail
1	2	2	
2	5	5	
3	4	4	
4	1	1	
5	3	3	EWR
6	6	12	EWR
7	10	10	
8	7	8	
9	12	13	JFK
10	9	7	JFK
11	11	6	
12	14	11	
13	8	9	
14	13	14	
15	15	15	
Dep. Runway Throughput	45 departures / hr	43.9024 departures / hr	
Cumulative Dep. Position Shifts	13	16	

Table 6.4: Example - “Aircraft to slot” assignments under Miles In Trail restrictions

These runway operations plans suggested under the two traffic scenarios are also depicted in Figure 6.9, in which the left side corresponds to the solution in the 2nd column of Table 6.4 and the right side corresponds to the solution in the 3rd column. The aircraft that are directly affected by the Miles In Trail restrictions are highlighted and crossings are not shown in the figure.

The active flow restrictions were Miles In Trail (MIT) separations for flights departing for two different destination airports:

- a) A 35-Miles In Trail separation requirement for all aircraft going to Newark airport (EWR):

assuming that one takeoff position difference translates to 5 miles of airborne separation once both aircraft are outside the terminal airspace and en route to their destination, a 35 Miles In Trail separation requirement was translated to a minimum takeoff distance of 7 positions between any two EWR-bound flights affected by the restriction, and

- b) A 30-Miles In Trail separation requirement (i.e. at least 6 takeoff positions distance) for all aircraft going to New York’s JFK airport (JFK).

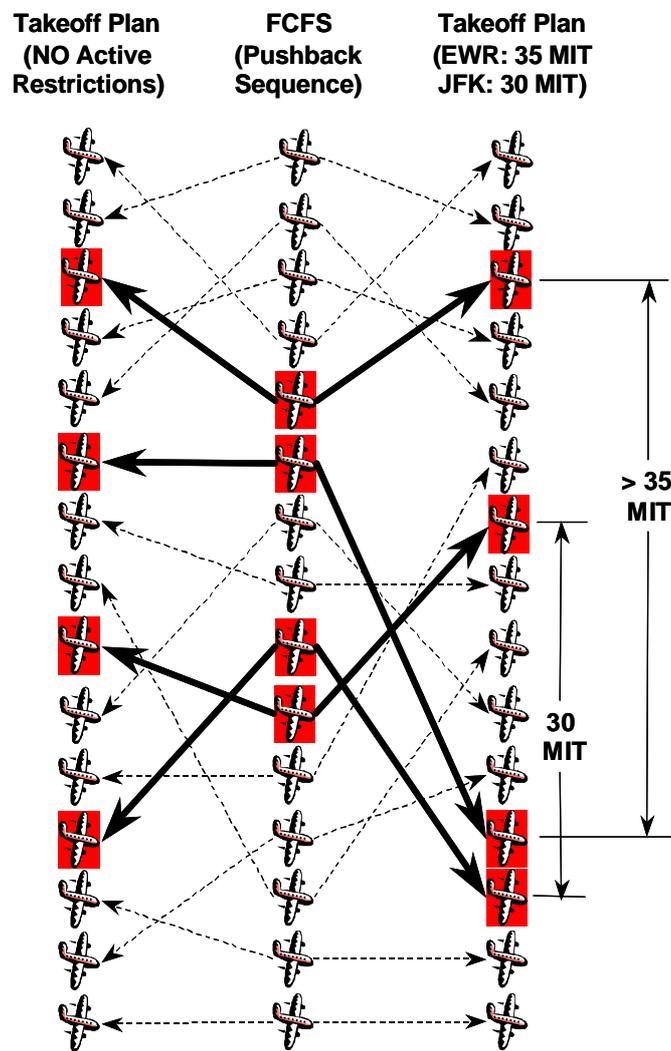


Figure 6.9: “Aircraft to Slot” assignments in an Unrestricted vs. a Restricted Traffic Scenario

If we compare the “aircraft to slot” assignments (takeoff plans) between the 2nd and 3rd column of Table 6.4 or between the left and right side of Figure 6.9, we can see that the differences in takeoff sequence positions for the same aircraft between the two runway operations plans were triggered by the presence of the flow restrictions. In other words, the two takeoff plans started having differences in the assignments of aircraft to takeoff slots as soon as the first pair of aircraft affected by flow restrictions was up for takeoff slot assignments.

The first restriction affects the 5th and 6th aircraft in the pushback sequence and the second restriction affects the 9th and 10th aircraft. In this particular example, when air traffic flow restrictions were not active (second column in Table 6.4, left side in Figure 6.9), both of these pairs of flights were optimally scheduled to take off 3 positions apart from each other (3rd and 6th for the first pair and 9th and 12th for the second pair). Once the restrictions became active (third column in Table 6.4, right side in Figure 6.9), such a takeoff plan was not acceptable. The first pair of flights needed to be at least 7 positions apart from each other and the second pair needed at least 6 takeoff positions between them. The final optimal takeoff plan resulted in the first pair of flights scheduled to take off 9 positions apart (3rd and 12th)³² and the second pair 6 positions apart (7th and 13th).

Flow restrictions seem to affect the departure throughput of the generated runway operations plan. There is a small throughput loss of about 1 departure / hour in the restricted case compared to the unrestricted one. Even though the two-stage planning algorithm attempts to maximize

³² More than the 7 positions necessary but scheduled for throughput and delay optimality over this entire set of 15 aircraft.

usage of available runway time, due to the new takeoff sequence mandated by the presence of flow restrictions and also due to the earliest times that each individual aircraft can be available at the runway the previously optimal weight class sequence is not feasible anymore and a less optimal weight class sequence has to be used. This results in slightly decreased runway usage compared to the runway plan in the unrestricted case.

Also in this example, by affecting only four (4) out of a total of fifteen (15) aircraft under planning, flow restrictions seem to increase the cumulative number of position shifts between the pushback and the takeoff sequence, as demonstrated in Table 6.4. The first of the EWR-bound aircraft still maintains its 3rd position in the takeoff sequence but the next EWR-bound aircraft has to be moved later in the sequence in order to satisfy the 35-Miles In Trail restriction. This takeoff position shift creates a “cascading” effect of takeoff slot re-assignments compared to the unrestricted scenario, which affects most of the subsequent aircraft. This results to a larger in the restricted scenario (16 total position shifts) vs. the unrestricted case (13 total position shifts).

As in the unrestricted scenario, performance benefits analysis of the two-stage algorithm in the presence of traffic flow restrictions was performed by following the testing procedure described in Figure 6.5 with the original pushback schedule of fifteen (15) aircraft used as the simulation input. Before the simulation results are presented, let us see what results could be expected based on our knowledge and understanding of airport ground operations. The First Come First Serve ground handling strategy is represented in Figure 6.10 as it was applied in the FCFS simulation, regardless of the pushback schedule input used (Original or Planned).

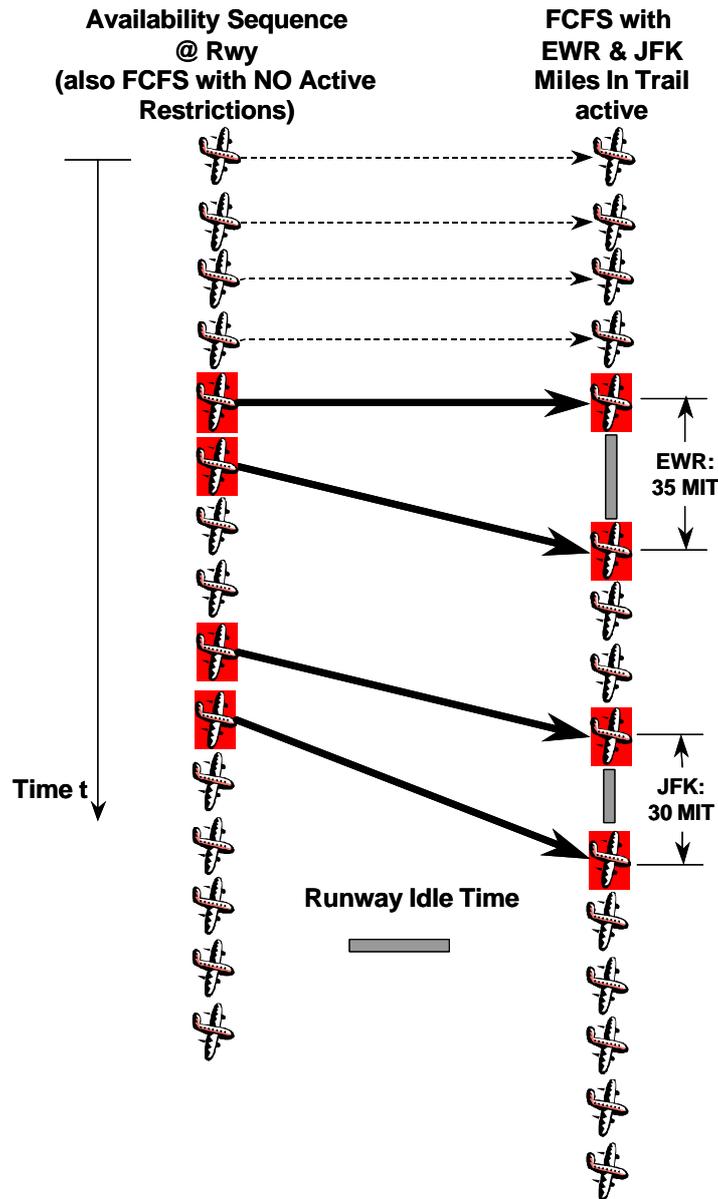


Figure 6.10: “Aircraft to slot” assignments with a First Come First Serve Aircraft Handling Strategy Under Miles In Trail Restrictions.

The left side corresponds to the sequence with which the 15 aircraft arrive at the runway queue and since a First Come First Serve strategy is used, this side also represents the takeoff sequence of the aircraft in a case where no flow restrictions are active and each aircraft takes off as soon as

the runway is clear from other departure and crossing operations. This left side serves as a reference to which the right side of the figure is compared. The latter corresponds to a case where Miles In Trail restrictions are active for both EWR and JFK airports. The two pairs of aircraft that are affected by the Miles In Trail restrictions are highlighted and crossings are not shown in the figure.

An important observation is that there is a penalty in the form of runway idle time to be “paid” if all flow constraints are to be satisfied. Comparing the right side of Figure 6.10 to its left side, the trailing aircraft of each aircraft pair that is affected by Miles In Trail restrictions absorbs a delay equal to the amount of runway idle time that is necessary in order to satisfy the Miles In Trail separation between the two aircraft. The runway idle time period necessary for satisfying the Miles In Trail restriction may not always correspond to total lack of activity on the departure runway. In some cases, runway crossings are scheduled in these time periods immediately after the first of a pair of Miles In Trail-restricted aircraft takes off. However, typically the time period necessary for satisfying the Miles In Trail separation is much larger than the amount of runway time needed for the crossing of a group of crossings that can fit in the taxiway space between the two parallel runways of the benchmark airport (Figure 5.1).

Each runway idle period is absorbed as a delay by all aircraft that take off after that period, in addition to any runway queuing delay that each of these aircraft may suffer due to other departure and crossing operations on the same runway. If only one flow restriction is active (as opposed to two), there will be less delay imposed in the takeoff sequence due to fewer runway idle time periods being introduced. The more restrictive the problem constraints become, the

more likely a runway idle time penalty is to occur and the larger the magnitude of that penalty is expected to be.

Based on these observations, the simulation test results are expected to indicate that:

- a) Active flow constraints have a diminishing effect on departure and total runway throughput, while at the same time they increase average departure queuing delays and elapsed times over all 15 involved aircraft, but however,
- b) Benefits from planning are expected to increase across all tested performance benefits dimensions. This is expected because, even though the flow restrictions create a traffic situation with higher inefficiencies (compared to the unrestricted case), the planner addresses those inefficiencies by including the flow restrictions in the constraint set of the optimization problem that is solved in the 2nd stage of the Runway Operations Planning algorithm. Therefore, the optimizing effect of the two-stage algorithm is expected to be more evident and lead to higher benefits than those experienced in the unrestricted traffic scenario.

In the FCFS traffic handling regime, the existence of active flow restrictions does not necessarily mean that the aircraft bound for the restricted destination airports are definitely affected. Due to the uncertainty in airport ground operations, even if two aircraft bound for a restricted destination airport push back from their gates close to each other, after their taxi process is complete they may reach the runway at a separation that already satisfies the restrictions. In that case, no introduction of runway idle time by the air traffic controllers will be necessary. Therefore, in order to ensure that there are enough cases in which EWR-bound and JFK-bound

aircraft were actually affected by the active flow restrictions, so that reliable statistical results can be drawn, 25,000 Monte Carlo runs were performed in all cases of restricted traffic scenarios.

For the same set of fifteen (15) aircraft that was used in the unrestricted traffic scenario, the benefits analysis results (for all 25,000 Monte Carlo runs) are summarized in Table 6.5 for all the performance metrics tested.

	Original Pushback Input	Planned Pushback Input (Final Queuing Curve)	BENEFITS	
			YES	NO
Dep. Runway Throughput (departures / hour)	34.73	38.74	≈ 4 departures/hour	
Total Runway Throughput (operations / hour)	85.64	88.87	3.23 operation/hour	
Cumulative Dep. Delay (min)	55.3	51.4	≈ 15 sec / aircraft	
Cumulative Elapsed Time (min)	243.8	239.9	≈ 15 sec / aircraft	

Table 6.5: Benefits Analysis for a Restricted Traffic Scenario - EWR & JFK are restricted airports

Comparing the planning performance under the restricted scenario (Table 6.5) with the planning performance under the unrestricted scenario (Table 6.2) seems to verify our earlier expectations. It is evident how, regardless of whether planning is performed or not, the presence of the Miles In Trail restrictions reduces departure and total throughput compared to the unrestricted scenario ($38.74 < 39.92$ departures/hour with planning and $34.73 < 37.44$ departures/hour without planning) and increases the departure delays ($51.4 > 48.89$ min with planning and $55.3 > 40.98$ min without planning) and the cumulative elapsed times ($239.9 > 237.3$ min with planning and $243.8 > 229.6$ min without planning). However, the difference in performance between the unplanned FCFS case and the case when planning is performed is larger in this restricted

scenario than in the case with no active flow restrictions, with the planned case demonstrating better throughput and delay performance. This means that the potential operational benefits increase (as expected) when flow restrictions are active.

The benefits analysis results are both for the restricted and the unrestricted scenarios are presented in Figure 6.11, Figure 6.12, Figure 6.13, and Figure 6.14, each of which corresponds to one of the throughput or time-based performance metrics tested. For each of the four performance metrics, when its standard deviation is considered together with its mean value, in most cases the confidence intervals for both the unrestricted and the restricted case overlap and therefore, it appears that the differences between the means from the scenario with planning and the scenario without planning are not statistically significant. However, ANOVA statistical tests were performed for each of the above metrics and the statistical significance of the difference in mean values between the planned and the unplanned scenario was confirmed with a confidence level of 95 %. In addition, as shown in Table F.1 in Appendix F, mean standard errors were also very low, which supports confidence in the statistical significance of the difference in mean values.

Without the presence of restrictions, even with planning, each aircraft needed to absorb a certain level of “essential delay” for the sake of maximizing departure and total throughput. The term “essential delay” is used here to describe the delay that aircraft suffer because of the demand pressure exercised on the departure runway in order to achieve a high level of departure throughput. Whatever additional delay aircraft may absorb can be called “gratuitous delay” and the purpose of planning is to minimize the amount of that gratuitous delay that aircraft may

suffer beyond the necessary level of essential delay.

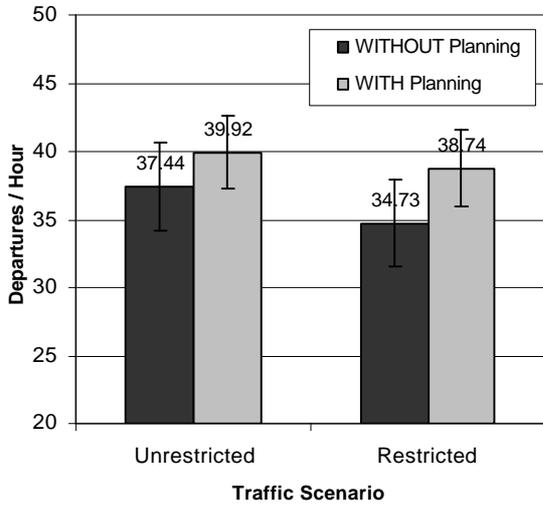


Figure 6.11: Departure Throughput

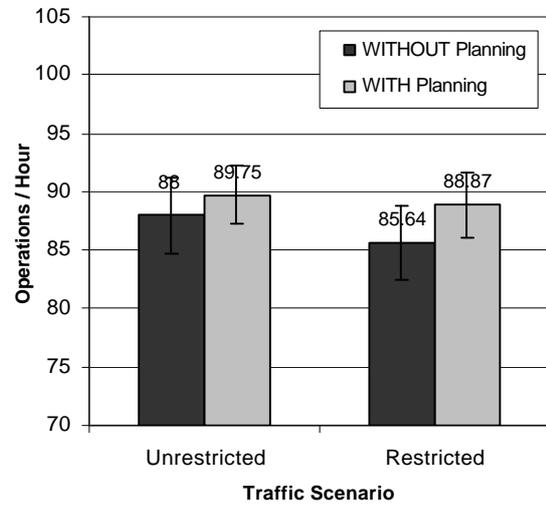


Figure 6.12: Total Throughput

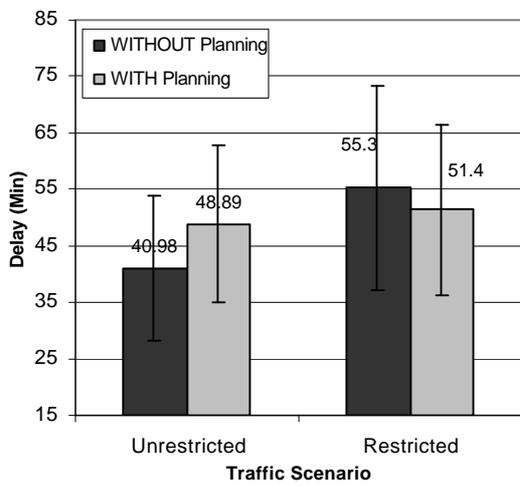


Figure 6.13: Cumulative Departure Delay

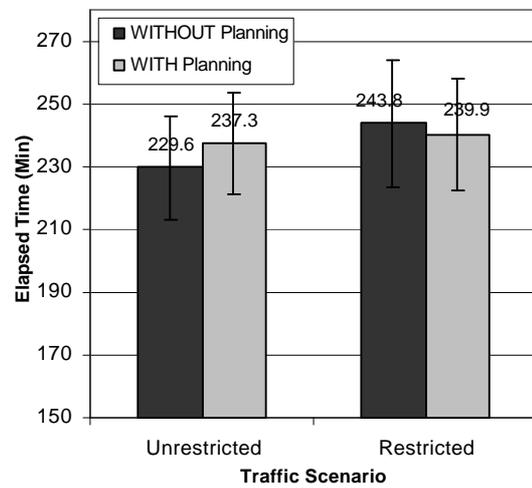


Figure 6.14: Cumulative Elapsed Time

As expected and as shown in Figure 6.13 and Figure 6.14, the presence of flow restrictions

introduces system inefficiencies that result in reduced departure and total throughput and gratuitous delays are more likely to be increased. However, planning seems to mitigate some of these adverse flow restriction effects by minimizing the gratuitous delay, which then seems to result in reversing the delay and elapsed time trend that manifested in the unrestricted scenario. In this restricted case, it turns out that, on average, planning increases departure and total throughput compared to the unplanned case, while at the same time reduces the total departure delays and elapsed times. This is achieved by taking into account the active flow restrictions early enough in the construction of the runway operations plan by including them in the constraint set of the optimization problem of the 2nd stage. Therefore, the generated runway operations and gate pushback plans schedule those aircraft that depart for restricted destinations to push back and takeoff in positions that satisfy the restriction separation, even though, at the same time, the set of aircraft being planned still absorbs “essential delay” at the runway in order to maintain a high departure throughput. This leads to actual delay and elapsed time benefits in the restricted scenario, as opposed to the small additional delay per aircraft that resulted from planning in the unrestricted traffic situation.

So, in summary, for this example restricted scenario, when runway operations planning was performed and the generated runway plan was then used as a guide for planning gate pushback operations, airport performance benefits manifested as follows:

- a) Departure and total operations throughput achieved on average higher values compared to the case when no runway operations planning was performed and
- b) Cumulative queuing delays and elapsed times were on average decreased.

The calculated elapsed time savings per aircraft reported above may seem very small. However, if they are used to produce a rough estimate of environmental benefits as a result of reduced emissions from aircraft engines, they may seem more valuable. The elapsed time savings were calculated based on the assumption that there is constant departure pressure on the runway, which is not true throughout a typical operational day at any given airport. Therefore, for estimating emissions and monetary savings, we assume that 50 % of the daily departure operations occur at busy times and the estimates reported correspond only to operations during busy periods. Since the benchmark airport used in the simulation (Figure 5.1) is representative of a runway configuration that is very common at Boston Logan airport, the following estimates are calculated based on operational data for that airport. The managing authority of Logan airport [69] reports that there are about 373,000 operations per year served at the airport, out of which we assume that 50 % are departures. Therefore, there are about 93,250 (= $0.5 \times 0.5 \times 373,000$) departures per year that occur at busy periods.

Using a methodology for calculating fuel flow and emissions based on Miller's work [91] and soliciting fuel flow coefficients and emission indices from [70], annual fuel savings for the airlines and annual reduction of emitted aircraft engine pollutants were estimated. Also, monetary values for estimating the value of pollution were taken from [93] and annual savings in pollution were calculated. The results are presented in Table 6.6.

Elapsed time savings can also be translated to benefits for airlines and their passengers in the form of a monetary value that can be attributed to operating cost savings for the airlines and travel time saved for the passengers. Monetary values for an average aircraft at Logan airport

(irrespective of weight class) were taken from [51] and it was estimated that the elapsed time savings from planning runway operations translate to 951,374 US \$ / year of combined savings in airline operating costs and passenger time value.

	SAVINGS	
	Kg / year	US \$ / year
Aircraft Fuel	396,578	142,768
Nitrogen Oxides (NO_x)	1,530	16,177
Hydrocarbons (HC)	1,849	7,045
Carbon Monoxide (CO)	10,180	814
Carbon Dioxide (CO₂)	1,251,203	37,536
TOTAL Pollution Cost Savings		61,572

Table 6.6: Estimated Annual Savings in Aircraft Fuel Burned and Reduction in Aircraft Engine Emissions (only BOS departures and only at periods of high traffic volume)

Because the results presented in Table 6.5 for the restricted scenario do not come purely from cases where the restrictions actually had an impact on the runway operations³³, they cannot fully demonstrate the effect of flow restrictions on throughput and delays. In order to further illustrate the effect of restrictions, the restricted traffic scenario is examined in more detail.

Figure 6.15, Figure 6.16, Figure 6.17 and Figure 6.18 present the results from scrutinizing this restricted traffic scenario. The scenario was run 25,000 times both with and without planning, among which there were:

- For the case WITHOUT planning: about 9,500 cases in which exactly one pair of aircraft was affected by one of the two active Miles In Trail restrictions (denoted as “Affected By One

³³ As it was mentioned earlier, there are always cases when, even though flow restrictions are active, no aircraft are affected by them and therefore system throughput and delays are not affected either.

MIT” in the figures) and about 14,000 cases in which exactly two aircraft pairs were affected by the two existing restrictions (denoted as “Affected By Two MIT” in the figures) and

- For the case WITH planning: about 9,500 cases in which exactly one pair of aircraft was affected by one of the two active Miles In Trail restrictions, but only about 2,300 cases in which exactly two aircraft pairs were affected by the restrictions. This number is lower than in the case without planning, because, as it was mentioned earlier, the introduction of the planner addresses the problem of satisfying flow restrictions early enough to prevent the occurrence of many cases where the restriction-affected aircraft will not arrive at the runway already at the desired Miles In Trail separation and will need to absorb additional delay, which translates to runway idle time, for the separation to be met.

In the remainder of the cases there was no aircraft that had to be delayed because of the active restrictions and therefore there was no runway idle time introduced in the system (denoted as “Affected By No MIT” in the figures).

Under the restricted traffic scenario, when the planner is used throughput benefits from planning runway and pushback operations are evident to the order of 2 to 3 departures / hour but, as restrictions become more severe and the number of aircraft pairs affected by flow restrictions increases, a gradual deterioration of departure and total throughput is demonstrated in Figure 6.15 and Figure 6.16.

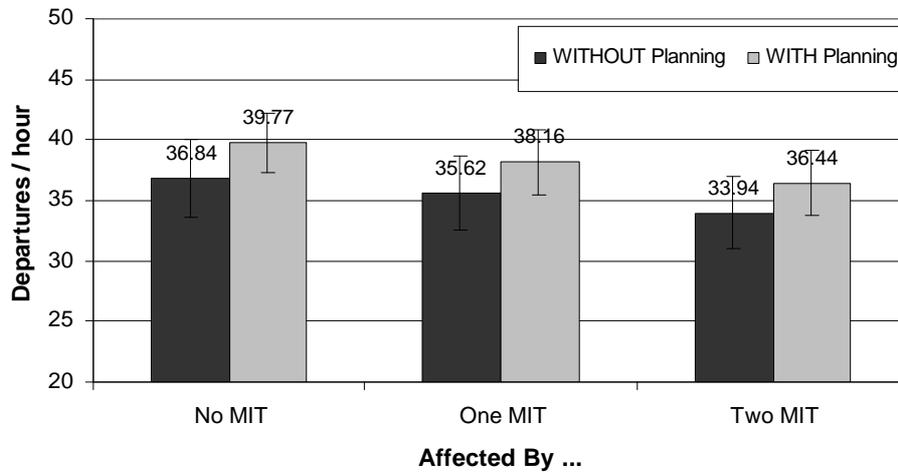


Figure 6.15: Departure Throughput

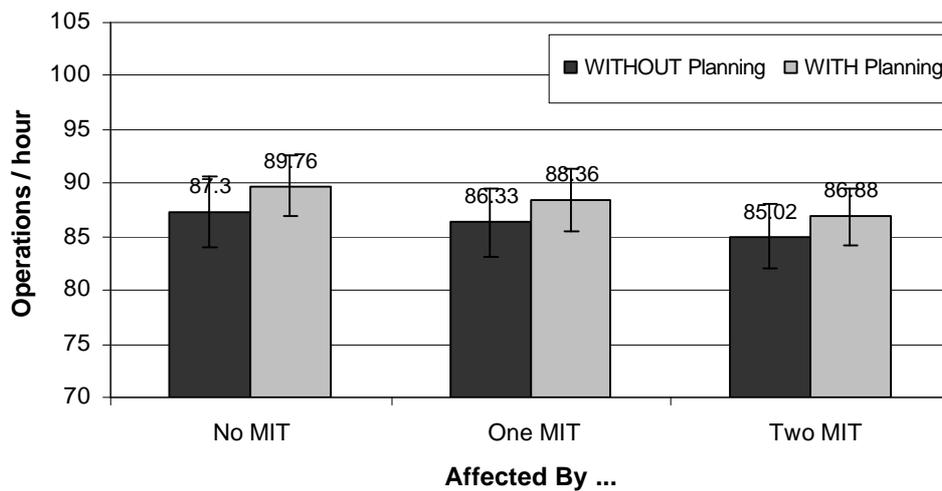


Figure 6.16: Total Throughput

Also, as it was expected, a gradual increase of departure delays and cumulative elapsed times is observed in Figure 6.17 and Figure 6.18. As restrictions affect the system more, the “gap” between delays with and without planning closes because the planning algorithm minimizes the gratuitous delay that the effect of restrictions attempts to introduce in the system.

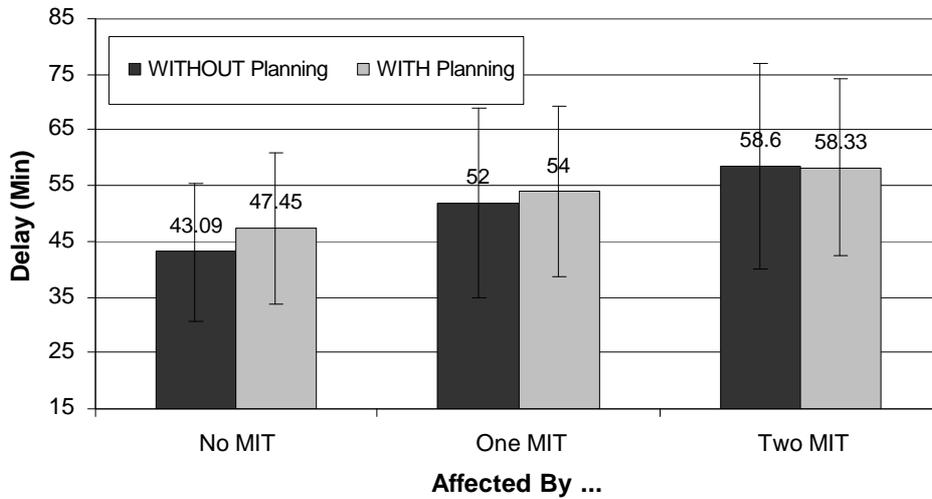


Figure 6.17: Cumulative Departure Delay

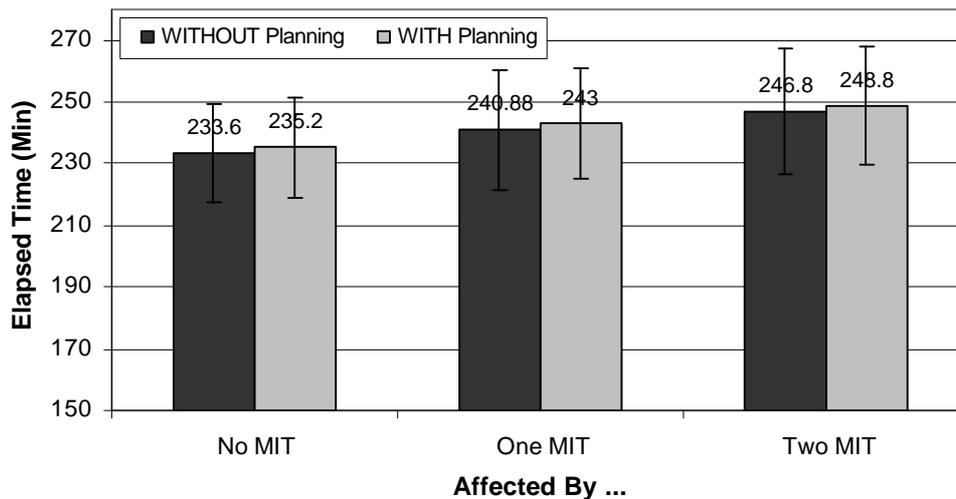


Figure 6.18: Cumulative Elapsed Time

When we focus on each of three subsets of cases in which: i) no aircraft pair, ii) one aircraft pair and iii) two aircraft pairs absorb runway delays due to the active flow restrictions, there seem to be no real planning benefits, but at least for departure delays, the planner seems to perform

progressively better as the number of restriction-affected aircraft pairs increases. So, despite the apparent absence of delay and elapsed time benefits, as presented in Figure 6.13 and Figure 6.14, when all three different cases of 0, 1 and 2 restriction-affected aircraft pairs are examined together (as they actually manifest in the real world), small delay and elapsed time planning benefits appear in the airport system. Standard error analysis for these results is presented in Table F.2.

Another type of air traffic flow constraints, commonly issued by ATC authorities, such as the ATC System Command Center in Washington or the en Route ATC centers, is time window restrictions³⁴. These are typically the result of a Departure Sequencing Program (DSP) or an Expected Departure Clearance Time (EDCT) slot constraint. Such a restriction is usually “translated” by air traffic controllers to an approximate window of permitted takeoff slots for the aircraft on which the restriction is imposed.

As far as benefits are concerned, the existence of such time window restrictions is expected to have an impact on throughput and delays similar to the impact of Miles In Trail restrictions. That is because in this case also, satisfying all time-window flow restrictions is achieved through the introduction of runway idle time in the FCFS simulation. This is demonstrated in Figure 6.19 through an example traffic scenario with multiple active air traffic flow restrictions. The sequence with which the 15 aircraft in Figure 6.19 arrive at the runway is portrayed on the left side of the figure. The right side corresponds to a scenario in which, a single Miles In Trail

³⁴ These types of restrictions were also discussed in section 3.8.1.

restriction to EWR is active and in addition, the flight that arrives 8th at the runway (framed aircraft in Figure 6.19) has a DSP restriction. The DSP time window corresponds to takeoff slots 10, 11 and 12 being the only ones permitted for that flight if all runway time was used (left side of Figure 6.19). However, since aircraft are handled on a First Come First Serve basis as they arrive at the runway and the DSP restricted aircraft arrives 8th at the runway, idle time needs to be introduced in the takeoff schedule, in order for the DSP restricted aircraft to takeoff within the permitted time window, as shown in the right side of Figure 6.19.

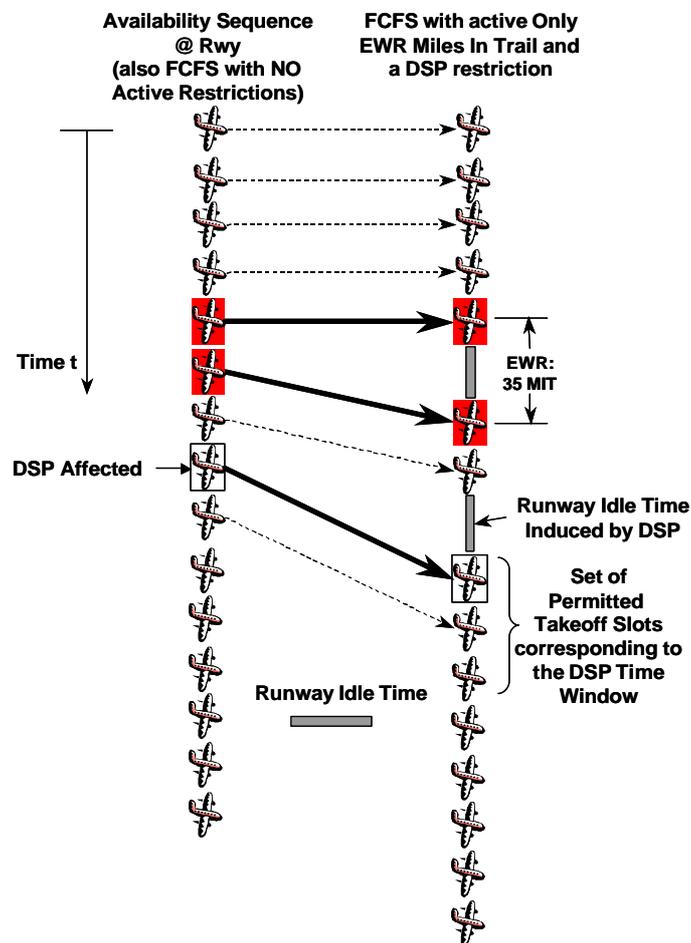


Figure 6.19: “Aircraft to slot” assignments with a First Come First Serve Aircraft Handling Strategy Under Miles In Trail and DSP Restrictions.

6.4 The Effect of Pushback Time Uncertainty on the Robustness of Runway Operations Plans

The emergence of benefits from planning runway operations with the decomposition-based two-stage algorithm, especially in traffic scenarios with active air traffic flow restrictions (section 6.3.4), make this algorithm a promising candidate for use in an airport ground traffic decision-aiding tool. However, the level of airport operational efficiency that can be achieved by using this algorithm, cannot alone determine its utility and applicability in a real-world operational environment. Another important criterion for the applicability of the algorithm and consequently of a decision-aiding tool that uses this algorithm, is the robustness of airport operational performance resulting from the tool's planning suggestions, in the presence of varying levels of uncertainty that can impact the pushback schedule input supplied to the decision-aiding tool.

Oftentimes, unexpected pushback events mandate modifications in gate and runway operations, because they change the exact timing when aircraft actually push back from their gates³⁵. In such an event, the level of robustness of solutions generated by the algorithm can determine how beneficial the algorithm can be in improving the airport's operational performance. So, in this section, we examine the robustness performance of the algorithm output with respect to operational metrics such as the departure throughput, departure delays and elapsed times resulting from the runway operations plans, in the face of unforeseen events that may alter the pushback schedule.

³⁵ ... which is what the algorithm used as an input.

Unforeseen events that affect the implementability of a runway operations plan can occur at any point in time, before or after air traffic controllers begin implementing the plan. For example, depending on when aircraft status information becomes available to airline dispatchers, the latter may communicate with tower controllers at any time to inform them that one or more aircraft will not be available to push back as planned. How early or late such information is available to the tower, can determine whether it is practical for a new runway plan to be generated and implemented or whether the old plan could still be used possibly with small modifications.

Consequently, in the context of this runway plan robustness analysis, there are various operational scenarios that can be examined, which pertain to the application of re-planning or not in the face of uncertainty in the pushback schedule. Before these operational scenarios are described, it is necessary to recall the following key points from the testing methodology that was used in the benefits analysis in sections 6.3.3 and 6.3.4 (Figure 6.5):

- An original (published) pushback schedule is the only starting point available to air traffic planners in their effort to plan airport operations by using the two-stage Runway Operations Planning algorithm,
- The operational value of any runway operations plan generated by the two-stage algorithm is easier to assess if the runway plan is “translated” back to the gates as a plan of pushback operations by using the back-propagation methodology presented in Chapter 4,
- Once a plan of aircraft pushback operations is created, due to the stochasticity of pushback operations it is not always certain that these pushback operations will materialize exactly as dictated by that plan,

- After pushing back from their gates according to the generated pushback plan, aircraft are handled through the taxi and takeoff process based solely on the air traffic controllers' operational experience, with no further planning guidance by any decision-aiding tool. So, in order to evaluate the throughput and delay performance that air traffic controllers can expect to achieve by using the planned pushback schedule that was created based on the runway planning suggestions of the two-stage algorithm, the pushback plan must be fed through a simulation of a simple First Come First Serve ground traffic handling strategy (FCFS simulation as it was introduced in section 6.3.1), which in most cases is representative of the ground traffic handling methods that controllers use at a real-world airport environment.

6.4.1 Methodology for Testing Robustness of Runway Operations Plans

Output plan robustness was tested using a Monte Carlo simulation methodology, as was done earlier for the benefits analysis. A pushback schedule consisting of fifteen (15) aircraft originating from two different terminals (as in Figure 5.1) was used as an input to the simulation. In order to remove any pushback timing effects³⁶, the pushback times of the 15 aircraft were evenly spaced at 60 sec apart from each other to create a regular pushback schedule, that will be referred to as the “nominal pushback schedule” or “nominal schedule” for the remainder of this text. The selected weight class mix for this nominal schedule was: Two (2) Heavies (13.33 %), ten (10) Large (66.67 %) and three (3) Small (20 %) aircraft, which is representative of the traffic mix encountered at many US airports. It was assumed that no flow restrictions are active.

³⁶ ... which will be examined in section 6.6

For representing a scenario with no pushback uncertainty, the nominal schedule was used as an input to the two-stage algorithm and a plan of runway operations was generated, which is referred to as the “nominal plan”. In most cases tested, some of the aircraft involved in the planning were actually assumed to push back on time. For the rest of the aircraft, random perturbations were introduced to their scheduled pushback times in the nominal schedule. These aircraft will be referred to as “aircraft that do NOT push back on time”. In a Monte Carlo fashion, several hundreds of pushback schedules were generated by introducing random perturbations to the pushback times of the nominal schedule, each of them was entered as input to the planning algorithm and the resulting runway operations plans were studied. The only aircraft characteristic that changed between runs was the pushback time. The destination, weight class category and originating terminal for each of the aircraft in the schedule were randomly assigned, but these characteristics were kept constant between different Monte Carlo runs.

Two different methods were used for generating varying levels of pushback schedule uncertainty, in order to test the effect of two specific parameters on the robustness of the output runway operations plans. These parameters, referred to as the “uncertainty parameters”, are:

1. *The number of departing “aircraft in the schedule that do not push back on time”*: In real-world operations it is highly unlikely that in a group of scheduled departures all of them will not be able to push back as planned. Therefore, in each Monte Carlo run, it was decided that a subset of the 15 aircraft in the schedule should have their pushback times altered in order to create a new schedule with perturbed pushback times. To test the effect that the number of aircraft that do not push back on time has on output robustness, simulation tests were conducted for different scenarios in which 2, 3, 4, 5 and 10 aircraft (out of 15) had their

pushback times changed within the same Monte Carlo run. In each run it was randomly selected which particular aircraft from the group would be the ones that do not push back on time, regardless of how many they were. For comparison reasons, cases where all 15 aircraft had their pushback times affected by uncertainty were also run.

2. *The value of the standard deviation of the pushback uncertainty distribution:* Analysis of historical data [94] determined that a lognormal distribution was the most appropriate distribution for aircraft pushback delays. The mean pushback delay time was set to 50 seconds. The standard deviation assumed different values within a range of 60 to 180 seconds, i.e. being equal to up to three times the time interval of 60 sec between consecutive aircraft pushback operations in the nominal schedule.

For our testing purposes the following three operational scenarios were used:

- a) *Scenario WITHOUT Planning:* In this scenario, graphically presented in Figure 6.20, the robustness of the original pushback schedule is investigated without any operations planning attempted. This is a baseline scenario that provides a comparison basis for the results from the remaining two scenarios. This is exactly the scenario WITHOUT Planning represented at the top part of Figure 6.4, with the only difference being that, in this case, the input to the FCFS simulation is an “Actual Pushback Schedule” that is generated by perturbing the pushback times of the original pushback schedule. Of course, as before, the FCFS simulation is involved in the testing process in an attempt to simulate ground traffic handling as it is typically done in a real world airport operational environment.

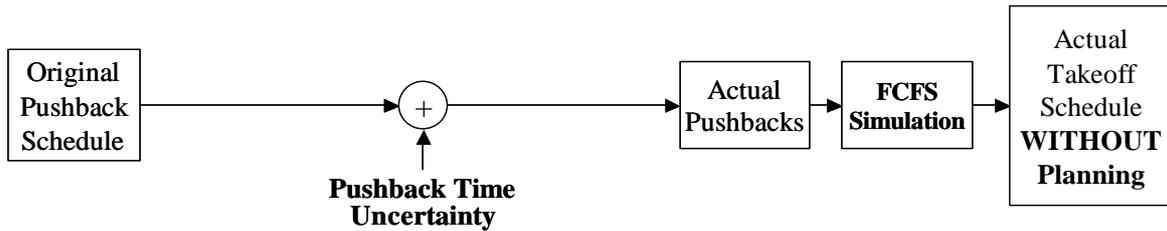


Figure 6.20: Robustness Analysis WITHOUT Planning

- b) *Scenario WITH Planning*: In this case, initially, the original pushback schedule is fed once without any uncertainty perturbations through the two-stage algorithm and a runway operations plan is generated. Based on the latter, a pushback plan is also created. It is the robustness of that pushback plan that is tested in this scenario by perturbing its planned pushback times. This scenario is presented in Figure 6.21. This is exactly the scenario WITH Planning represented at the bottom of Figure 6.4, with the only difference being that, in this case, the input to the FCFS simulation is an “Actual Pushback Schedule” that is generated by perturbing the pushback times of the planned pushback schedule (Pushback PLAN in Figure 6.21). It can be assumed that such an operational scenario corresponds to cases when pushback time changes occur late enough to make it impractical to generate a new runway plan and that is why runway operations planning is attempted only once in the beginning.

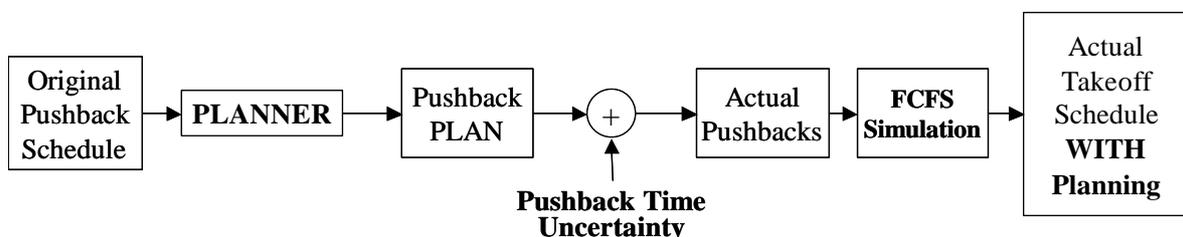


Figure 6.21: Robustness Analysis WITH Planning

c) *Scenario WITH ADAPTIVE Planning*: This scenario is similar to scenario (b) in the sense that the runway plan that was initially generated is rendered unimplementable because of pushback time changes. However, this scenario corresponds to cases when changes to the pushback times of certain aircraft occur with enough advance notice to have time for a new runway operations plan to be generated and used as a replacement. Therefore, a runway operations plan is generated more than once under this operational scenario. As shown in Figure 6.22, pushback uncertainty acts on the pushback times of the original pushback schedule and every time this happens an actual pushback schedule is generated. This actual schedule is then used as an input to the planner for a new pushback plan to be generated.

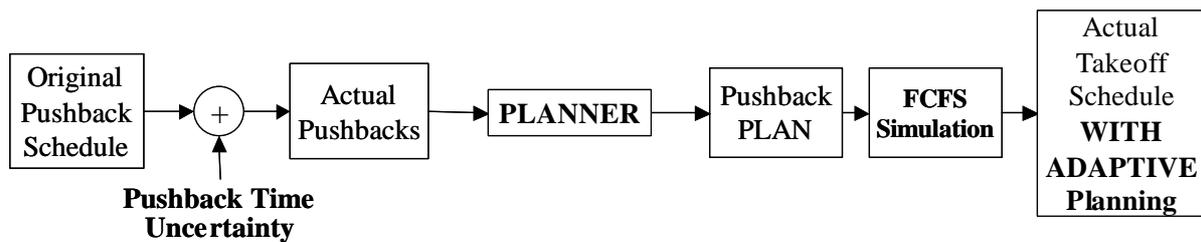


Figure 6.22: Robustness Analysis WITH ADAPTIVE Planning

Within each of the above operational scenarios, every new set of values for the uncertainty parameters constitutes a new simulation case. Each of the three scenarios can include many different simulation cases. For each simulation case, 500 Monte Carlo runs were performed³⁷, i.e. 500 different pushback schedules were generated with pushback times impacted (changed) by uncertainty. For each simulation case, the operational data resulting from the Monte Carlo runs produced statistical values for operational metrics, such as departure throughput and delays.

³⁷ Pertaining to this robustness analysis, as well as to the stability analysis presented in the next section, a short discussion on the convergence of the Monte Carlo simulation, which explains how it was found that 500 runs for each simulation case were sufficient, can be found in Appendix E.

6.4.2 Robustness Results

The above three operational scenarios were run as described in a Monte Carlo setting for different simulation cases. Each simulation case that was run corresponded to a different set of values for the two uncertainty parameters: i) the number of aircraft that do not push back on time and ii) the standard deviation of the uncertainty that impacts their pushback times. Sample results from the scenario WITH Planning (Figure 6.21), for various numbers of aircraft that do not push back on time and for three different values of uncertainty standard deviation are shown in Figure 6.23 for the departure throughput and in Figure 6.24 for the departure delays. Robustness behavior with respect to total throughput and elapsed time is also very similar. In each case, the results for zero “aircraft that do not push back on time” correspond to the throughput and delay resulting from the nominal runway operations plan generated by the planning algorithm for the nominal pushback schedule input. i.e. evenly-spaced pushback times every 60 sec without any uncertainty affecting the pushback times of any aircraft.

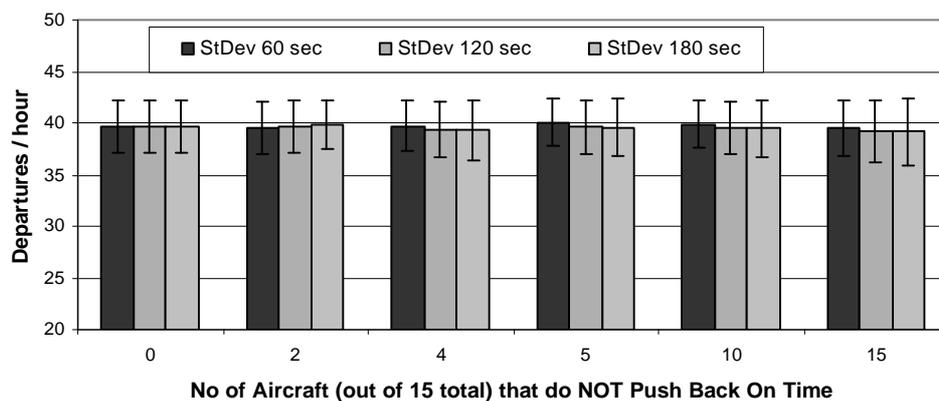


Figure 6.23: Robustness of Departure Throughput WITH Planning

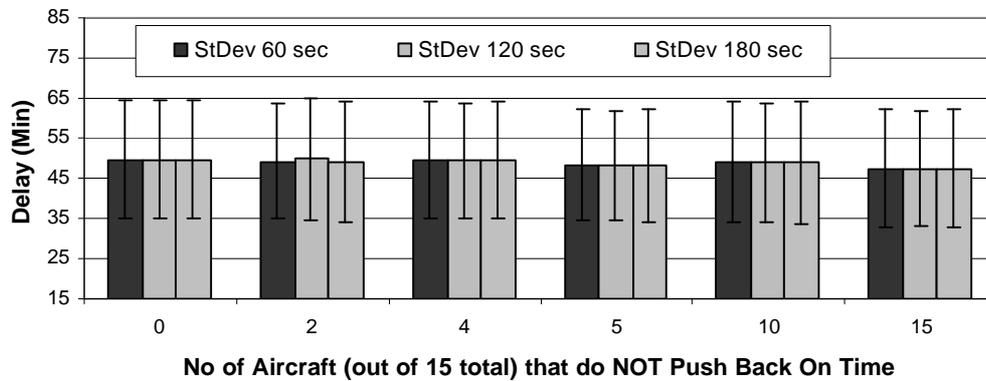


Figure 6.24: Robustness of Departure Delays WITH Planning

In the face of uncertainty impacting the planned pushback times generated by the planner in Figure 6.21, intuitively one would expect that, as uncertainty increases in severity and impacts a larger subset of the group of aircraft whose pushback times have been planned, a certain amount of the average throughput performance achieved by planning (right end of Figure 6.21) will be lost, because aircraft do not follow the optimally planned pushback operations any more. However, despite the impact of uncertainty, planning seems to lead to a fairly robust throughput (Figure 6.23) and delay (Figure 6.24) behavior across a wide range of uncertainty levels. This means that, when generating a new pushback plan is not an option for replacing a previous plan that was affected by unforeseen events (pushback uncertainty), the resulting throughput and delays should be expected to remain on average unaffected.

A robust throughput and delay performance also manifested in simulation tests that were conducted based on the operational scenario WITHOUT Planning (Figure 6.20). This means that even if no planning is involved in the departure process, the existence of varying levels of

pushback uncertainty impacting the pushback times in the original pushback schedule does not significantly change the average throughput and delay performance of the departure flow.

Given this, one would then wonder what the real merit emanating from the introduction of planning in the departure flow is. Recall that, as described in section 6.3, the primary benefit from planning departure operations comes in the form of higher departure throughput and in most cases lower departure delays compared to the airport performance achieved in the absence of planning. Furthermore, another important benefit of using the proposed method for planning runway and gate operations is that, as results show, the throughput and delay benefits experienced in the absence of uncertainty (section 6.3), are not compromised by pushback uncertainty regardless of the severity of its impact. This is demonstrated in Figure 6.25 and Figure 6.26 for a wide range of uncertainty parameter values.

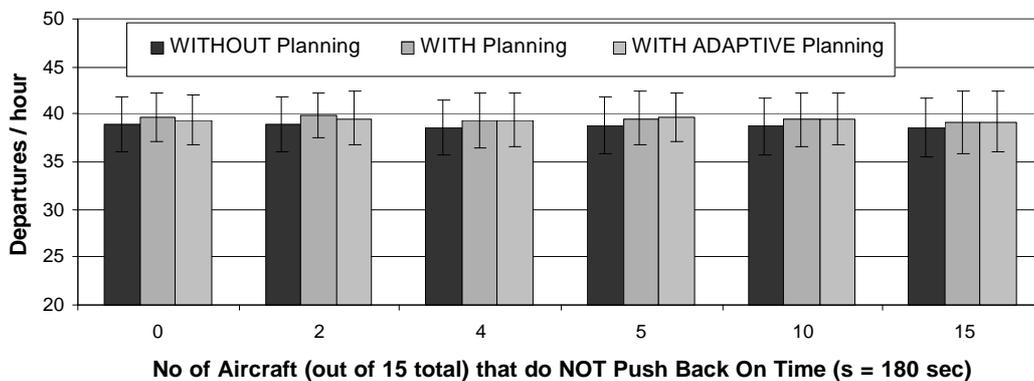


Figure 6.25: Departure Throughput Robustness Analysis

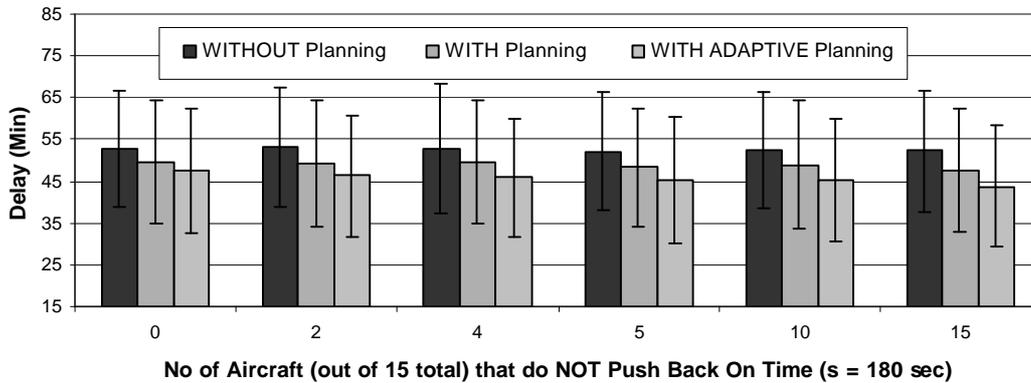


Figure 6.26: Departure Delays Robustness Analysis

6.4.3 The Effect of ADAPTIVE Planning

Initially it was established that planning yield operational benefits for the example examined in section 6.3 and then, the robustness analysis in section 6.4.2 showed that such benefits remain even when severe uncertainty impacts aircraft pushback times. Here, it is assumed that pushback uncertainty causes disruptions to a particular pushback plan and renders it unimplementable at a time early enough to allow for a new runway operations plan to be generated and used as a replacement of the “old” disrupted one. Under this assumption, simulation tests based on the operational scenario WITH ADAPTIVE Planning were also conducted and the corresponding departure throughput and departure delay results are also shown in Figure 6.25 and Figure 6.26.

In Figure 6.25, it appears that, when re-planning is performed as a response to pushback time uncertainty and unforeseen events that impact the planned pushback schedule, average throughput performance does not improve but does not deteriorate either. This is true regardless

of the level of uncertainty existing in the system (standard deviation value) and of the extent of the impact that this uncertainty has on the pushback schedule (number of aircraft that do not push back on time). Since the planner prioritizes departure throughput maximization, generating a runway and gate operations plan once (no re-planning), seems to be enough to ensure that at least throughput performance is maintained at a high level and in that sense, there seems to be no room for re-planning to improve throughput performance even more.

Furthermore, as it was seen earlier in section 6.3, planning aircraft to push from their gates earlier in order to exercise departure pressure on the runway and therefore enhance departure throughput, usually leads to departure delays and taxi out times that are higher than they would normally be if no planning had been performed. However, re-planning results for departure delay (Figure 6.26) indicate that there are moderate benefits with respect to departure delays that can be achieved through re-planning in the face of pushback uncertainty.

In summary, departure throughput may have proven difficult to further improve through re-planning, because it had already been improved significantly with the runway and gate operations plans that were originally generated. However, only with planning but without re-planning, due to the tradeoff between throughput maximization and delay minimization, prioritizing the improvement of departure throughput seems to have not allowed reduction of delays as much as feasible. That was revealed when re-planning was performed and simulation results indicated that, on average departure delays were further reduced through re-planning, without compromising average departure throughput.

6.5 The Effect of Pushback Time Uncertainty on the Stability of Runway Operations Plans

Human operators, e.g. air traffic controllers, traffic management coordinators and tower supervisors in an air traffic control tower, are the ones that are ultimately expected to use any type of airport ground traffic decision-aiding tool that utilizes the two-stage runway operations planning algorithm. In the face of unexpected pushback events, the output of the algorithm and therefore the planning suggestions of the decision-aiding tool can change significantly. Consequently, the level of stability of these runway operations plans generated by the algorithm may affect the amount of work the controllers will have to do in order to implement new plans, if necessary. Therefore, an important criterion for the usability of the decision-aiding tool in a dynamic airport environment is the stability of its output of planning suggestions in the presence of varying levels of uncertainty affecting the original or planned pushback schedule that is given as an input to the runway operations planning algorithm within the decision-aiding tool.

For example, assume that a runway operations plan has been generated and the appropriate tower controllers are in the process of implementing it. If one or more of the aircraft in the plan does not pushback in time as expected, then the current runway operations plan becomes unimplementable. At that point, controllers have the option to either continue with the existing runway operations plan after removing any “problematic” aircraft without replacing them with others, or to solicit another runway operations planning suggestion from the decision-aiding system. Of course, the usefulness of soliciting a new suggestion from the decision-aiding system

depends on how early or late is known to the controllers that certain aircraft are or will be unavailable to participate in the implementation of the original plan.

The first option of omitting the “problematic” aircraft does not incur any additional workload on controllers but results in some runway time being wasted. On the other hand, if the second option of generating a new runway operation plan is chosen, the stability of the algorithm’s output determines how different the new planning suggestion will be from the previous one in terms of the weight class sequence and final aircraft takeoff sequence and timing. Thus, the stability of the output affects how much additional effort the controllers will have to go through in order to modify the previous takeoff sequence and timing and implement the new one instead.

6.5.1 Methodology

In all the operational scenarios examined in this stability analysis, the starting (published) pushback schedule is the same “Original Pushback Schedule” with the same weight class mix that was used for the robustness analysis. In addition, any effect of the pushback timing of the starting pushback schedule is removed again with evenly spaced pushback times. The two “uncertainty parameters” whose values determine the scope of impact and the severity of pushback uncertainty are the same ones used in the robustness analysis, i.e. the number of scheduled departures that experience a pushback time change and the standard deviation of the pushback delay distribution.

6.5.2 Stability Metrics

To measure the stability of the algorithm's solutions, the following two metrics were used:

1. The frequency of occurrence (% of times among all Monte Carlo runs) of the event “the weight class sequence used to generate the optimal runway operations plan in a Monte Carlo run under a new input pushback schedule is identical to the weight class sequence used for the optimal solution in the nominal case (original input pushback schedule)” and
2. The frequency of occurrence (% of times among all Monte Carlo runs) of the event “the optimal aircraft (tail number) sequence in a Monte Carlo run under a new input pushback schedule is identical to the aircraft (tail number) sequence used for the optimal solution in the nominal case (original input pushback schedule)”.

The results presented below are from the same family of simulation cases that was run for the robustness analysis under the operational scenario WITH ADAPTIVE Planning, where each case corresponds to a particular combination of uncertainty parameter values for the number of aircraft that do not push back on time and the pushback delay standard deviation.

6.5.3 Stability with Respect to Weight Class and Aircraft Sequences

For a fixed value of 180 seconds for the pushback delay standard deviation, the two stability metrics are presented in Figure 6.27 as a function of the number of aircraft (out of a total of 15) that had their pushback times affected by uncertainty.

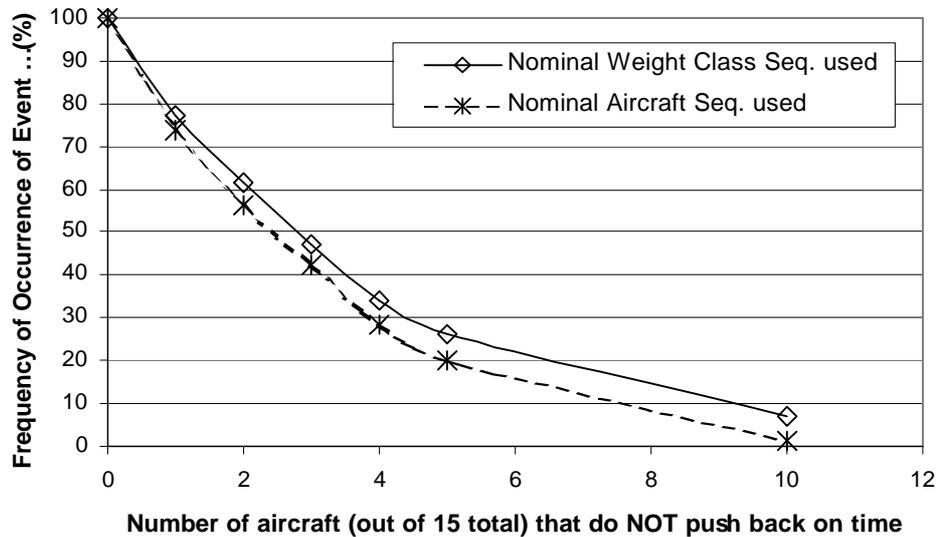


Figure 6.27: Stability of Algorithm Solutions - Pushback Uncertainty Standard Deviation of 180 sec

Each data point in this figure corresponds to a particular set of values for the two "uncertainty parameters". In all cases, zero (0) aircraft that do not push back on time means no deviation from the nominal case and translates to a value equal to 100 % for both metrics. In other cases, the number of times the event in stability metric 2 occurs (denoted as "Nominal AS (Aircraft Sequence) Used" in Figure 6.27) was lower than the number of times in which the event in stability metric 1 occurs (denoted as "Nominal CS (Class Sequence) Used" in Figure 6.27).

Measuring the value of the two "frequency of occurrence" metrics was achieved by counting the number of Monte Carlo runs (out of a total of 500 in each simulation scenario) in which the weight class sequence (aircraft sequence) used in the optimal solution was the same one as the weight class sequence (aircraft sequence) used in the optimal solution of the nominal case,

without considering the positioning of crossings in the sequences even though crossings were present. Crossings were not considered because, due to the different pushback times of the departing aircraft from one Monte Carlo run to another, crossings are likely to be serviced at different times in each run, even though, regarding only departures, the weight class and the aircraft (tail number) sequences of each Monte Carlo run may stay unchanged and possibly identical to those of the nominal case.

As was intuitively expected, the simulation results in Figure 6.27 provide evidence that the algorithm appears to be more stable in the face of uncertainty when there is a low number of aircraft in the entire pushback schedule that do not push back on time. Assume that a ground traffic decision-aiding tool utilizes the two-stage runway operations planning algorithm. Then, for example, it seems that if only one (1) aircraft within this particular set of 15 scheduled aircraft had a pushback time that does not materialize as expected, then the air traffic controllers and traffic management coordinators soliciting planning suggestions from the decision-aiding tool should expect the following:

- a) A chance of about 80 % that the two-stage algorithm, given a new pushback schedule as an input, will suggest a new runway operations plan with a weight class sequence identical to the weight class sequence of the original runway operations plan that was suggested before the pushback schedule input was altered. This means that, in planning runway operations, if the air traffic controllers solicit from the decision-aiding tool only a weight class sequence suggestion and use it to construct an aircraft (tail number) sequence as a runway plan without solving the 2nd stage of the planning algorithm, then they can still be 80 % certain that, even if the pushback time of one of the aircraft does not materialize as planned, the suggested

weight class sequence can still be used for planning runway operations.

- b) A chance of about 70 % that the two-stage algorithm, given a new input, will suggest a runway operations plan with an aircraft sequence identical to the original aircraft sequence that was suggested before the pushback schedule input was altered. In such a case, the air traffic controllers will need to introduce no changes to the runway operations plan even though one of the aircraft did not pushback from its gate as expected.

As expected, simulation results also show that, as the number of aircraft in the pushback schedule that do not push back on time increases to two (2) or more, the stability of the algorithm output deteriorates. More specifically, if there are more than two (2) aircraft that do not push back on time, there is only about 40 % chance (or less, depending on how many aircraft do not push back on time) that the original weight class sequence will still be optimal and only about 25 to 35 % chance (or less) that the original aircraft takeoff sequence will not need to be altered by the controllers.

Given the above results, we assume that, for up to two (2) aircraft not pushing back from their gates on time, the two-stage algorithm demonstrates acceptable stability. Based on this assumption, we now examine what is the effect that the level of severity of the uncertainty in the pushback schedule has on the stability of solutions suggested by the two-stage algorithm. The value of the standard deviation of the pushback delay distribution was varied within a range of 60 to 180 seconds (for a mean value of 50 seconds). The results presented in Figure 6.28 and Figure 6.29 are quite encouraging.

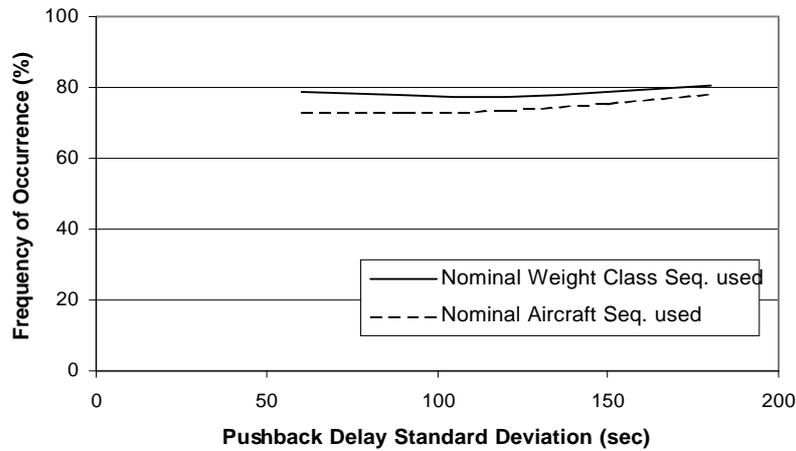


Figure 6.28: Stability of Algorithm Solutions when One (1) Aircraft has a Changed Pushback Time

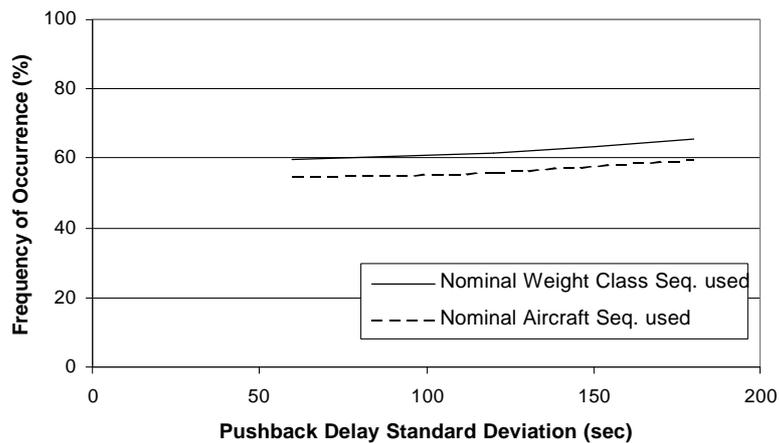


Figure 6.29: Stability of Algorithm Solutions when Two (2) Aircraft have Changed Pushback Times

It appears that, the runway operations plans manifest the same stability behavior across different levels of uncertainty introduced in the pushback schedule. In fact, that is why stability results as a function of the number of aircraft that do not push back on time were only presented in Figure 6.27 for one value (120 seconds) of the pushback delay standard deviation. The shape and

values of the two stability curves in Figure 6.27 do not change significantly for other standard deviation values.

Summarizing, it was established that, when uncertainty alters scheduled or planned pushback times, the output of the algorithm tends to be quite unstable in terms of the weight class and aircraft sequences suggested by the algorithm as optimal runway operations plans. For example, the planning algorithm seems to be much less stable when five aircraft (in a pushback schedule of 15 aircraft in total) have their pushback times affected by uncertainty, compared to the case when only one aircraft is affected. This means that, for an increasing number of aircraft that have their planned pushback times impacted by uncertainty, even though the algorithm output demonstrates a robust behavior with respect to the performance of the runway operations plans, throughput and delay optimality in these plans is not always achieved by means of the original (nominal) weight class and aircraft sequence.

Furthermore, it seems that, regardless of how high or low the pushback uncertainty is, the stability behavior of the algorithm depends only on one of the two uncertainty parameters: the number of aircraft that do not push back on time and does not depend on the standard deviation of the uncertainty. This means that, based on the stability analysis results for the particular regular schedule of 15 aircraft, if the number of aircraft that do not push back on time is somehow kept constant during real-world operations, then the two-stage algorithm can be expected to not lose or gain any stability performance as various local or remote events change the level of uncertainty existing in airport operations.

6.6 The Effect of Pushback Timing

As it was explained in Chapter 4, a queuing adjustment is introduced when the runway operations plan is back-propagated to a gate operations plan, in order to ensure that aircraft are planned to push back from their gates early enough to achieve adequate departure pressure on the runway.

However at the same time, the back-propagation method must ensure that aircraft do not push back very early to the point that, even though no significant throughput is gained by such early pushback operations, aircraft end up taxiing out for a long time or absorbing delays waiting at the runway takeoff queue.

So, when planning is introduced in departure operations, attention to such issues is necessary in order to prevent compromising departure delay performance for the sake of departure throughput and vice versa. This throughput and delay performance achieved with the introduction of planning is largely determined by the exact timing of pushback operations in the pushback plan designed by the planner, which in turn is strongly coupled with the expected timing of pushback operations in the original pushback schedule. Therefore, the timing of pushback operations in the original pushback schedule can have an effect on the ability of the planner to improve the performance of departures at an airport. For this reason, in this section, we isolate and test these effects of schedule pushback timing by investigating how beneficial the use of planning with the two-stage runway operations planning algorithm can be for several different schedule inputs. Six

pushback schedules were tested, each of which only differed from the rest in the distribution of pushback times, as depicted in Figure 6.30. The same number of 15 aircraft and the same fleet mix of 13.33 % Heavies (2 aircraft), 66.67 % Large (10 aircraft) and 20 % Smalls (3 aircraft) was used in all schedules.

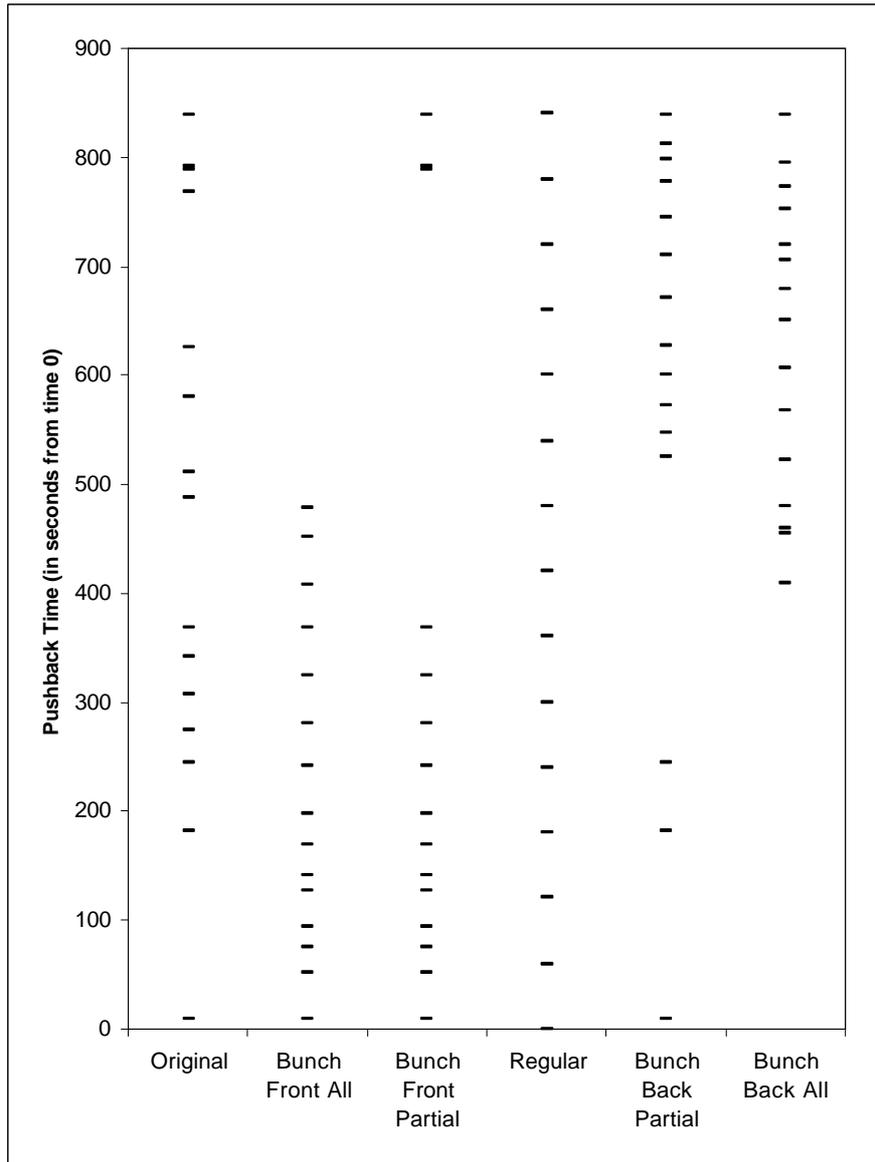


Figure 6.30: Pushback Schedules with Different Pushback Time Distributions

The schedules labeled “original” and “regular” are the only two that have already been tested as inputs to the planner. The original schedule is the one that was extracted from Logan airport data (section 6.3.2) and the regular schedule is the one with evenly spaced pushback times that was used in the study of pushback uncertainty effects in sections 6.4 and 6.5. The labels of the other four schedules in Figure 6.30 describe how each of them was created using the regular schedule as a basis.

For example, the “bunch front partial” schedule has most of the 15 aircraft scheduled to push back very close to each other at the front end of the time window [0, 850] that the pushback operations in the entire schedule are expected to occupy. After the initial “bunch” of aircraft pushback operations, a large gap of no pushback activity follows before the last three pushback operations are scheduled. The “bunch back partial” schedule is similarly constructed but to the back end of the pushback time window. The other two schedules (“bunch front all” and “bunch back all”) have all pushback operations scheduled close to each other, either at the beginning or at the end of the pushback time window. Benefits analysis was performed for each of these six schedules using the testing methodology that was introduced in section 6.3.2. The results of these benefits analyses are presented in Figure 6.31, Figure 6.32 and Figure 6.33³⁸ and standard error analyses can be found in Table F.3 and Table F.4.

³⁸ Total throughput manifests the same behavior as departure throughput in Figure 6.31 and for brevity reasons they are not presented.

6.6.1 Scenarios WITHOUT Planning

Before planning benefits results are discussed, it is interesting to observe the throughput and delay trends for the cases when no planning is performed. As expected, pushback timing has a significant effect on average departure throughput. The original pushback schedule leads to a certain level of average departure throughput (37.44 departures / hour), but as soon as pushback times are more closely scheduled to the front end of the time window, departure throughput is increased due to higher pressure exercised on the departure runway. This result supports the discussion presented in Chapter 4 about the relation between increased departure throughput and scheduling aircraft to push back from their gates earlier rather than later.

In addition, it seems that the location of a gap in the pushback activity in the schedule is an important determinant of departure throughput. The later such a gap exists, the higher the expected departure throughput (“Bunch Front Partial” vs. “Bunch Back Partial” in Figure 6.31).

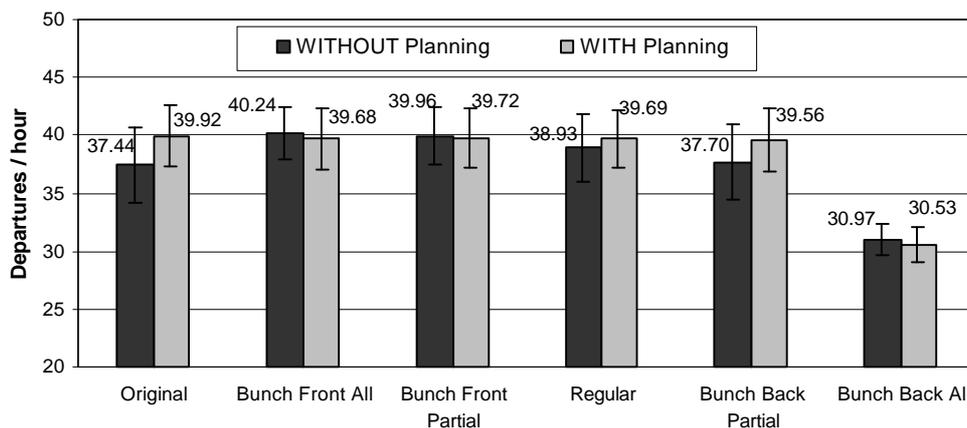


Figure 6.31: Effect of Pushback Timing on Departure Throughput

At the other end of the “pushback timing” spectrum, when pushback operations are scheduled closer to the back end of the time window, departure throughput is reduced because aircraft do not exercise as much pressure on the departure runway as before. Of course, a certain level of departure throughput is still maintained, but only until all aircraft are scheduled to push back later (“bunch back all” schedule), in which case the departure throughput is significantly decreased (30.97 departures / hour). Note that, in the “bunch back all” schedule the starting time point of the first pushback operation occurs much later than the first operation in any other case and all other pushback operations are closely spaced after that. Therefore, one would expect this case to manifest a high throughput also, due to high departure pressure on the runways. However, in order to maintain a common calculation basis among all six schedules that were tested, departure throughput in the “bunch back all” case was calculated by using as a starting time point of all operations the same early starting point (close to time 0 in Figure 6.30) that was used in the throughput calculations for all other schedules. Had that not been done, the departure throughput value corresponding to the “bunch back all” schedule would be much higher (about 40 departures / hour).

As expected, departure delays and elapsed times are increased compared to the original schedule when all or the majority of aircraft are scheduled to push back very early or very late, as is the case in all schedules except the original and the regular (Figure 6.32 and Figure 6.33).

When pushback operations are closely scheduled to occur early, aircraft are expected to spend more time taxiing since they leave their gates earlier and more time waiting at the runway queue since they are likely to join it earlier. When pushback operations are closely scheduled but later

in the schedule window, delays and elapsed times are also expected to increase due to the increased pressure on the runway, which results to longer runway queues. However, there is one case that provides an interesting observation.

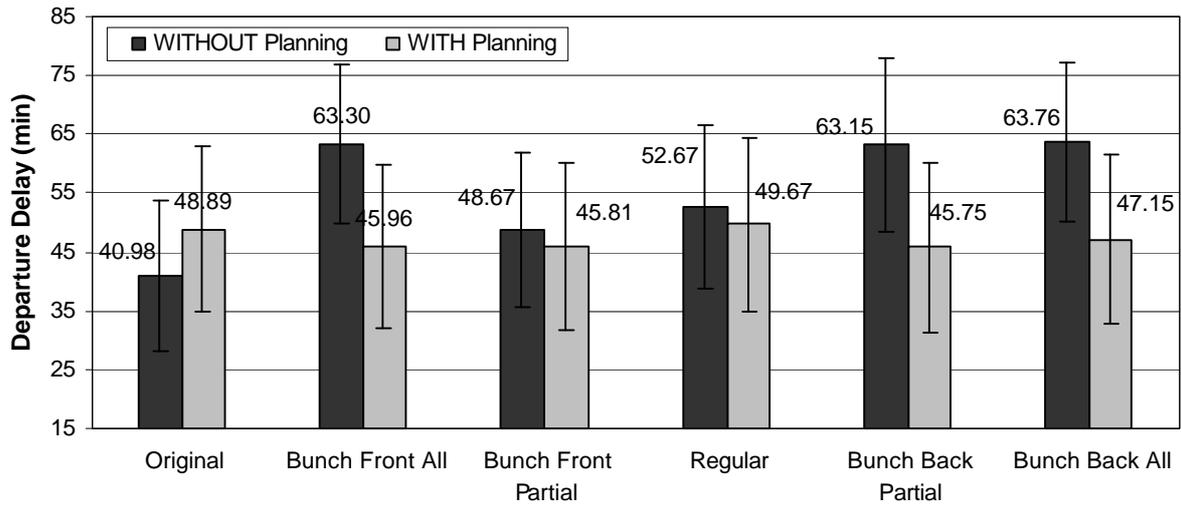


Figure 6.32: Effect of Pushback Timing on Departure Delay

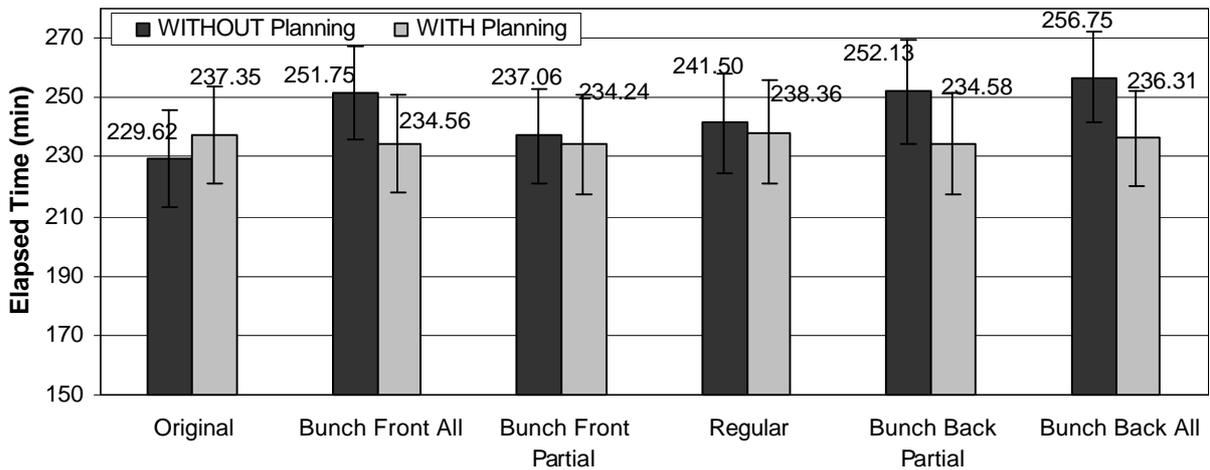


Figure 6.33: Effect of Pushback Timing on Elapsed Time

The location of a large gap in the pushback schedule can possibly have a diminishing effect on cumulative departure delays and elapsed times. For example, both the “bunch front partial” and the “bunch back partial” schedules include a large gap. The “bunch front partial” case assumes that this large period of no pushback activity is close to the end of the schedule. Even though one would expect delays in this case to be high due to the early aircraft “bunching”, nevertheless this schedule leads to lower delays than the “bunch back partial” schedule. That is because, the location of the pushback gap at the end of the schedule helps in reducing the cumulative delays by removing some taxi time and runway queuing time from the last few aircraft that push back after the gap in the schedule. These aircraft, which are usually the ones that absorb most delay among the entire schedule, are now expected to experience shorter taxi times and less time waiting at the runway queue (if any). This drives the cumulative departure delay and elapsed time to lower values than those for the “bunch back partial” or even for the “bunch front all” schedule. Had delay calculations stopped right before the pushback gap in the schedule, results would be different.

6.6.2 Scenarios WITH Planning

The distribution of pushback times in the original pushback schedule was shown to affect the level of operational performance that can be reached without any departure planning involved. However, even when planning is performed, pushback timing seems to be coupled to the degree of effectiveness of the planning process.

As shown in Figure 6.31, the planner is able to improve average departure throughput only in

cases where, in the absence of planning, the timing in the original pushback schedule is not such that leads to very high or very low average departure throughput values to begin with. For example, in the cases of the “Bunch Front All” and the “Bunch Front Partial” schedules, the concentration of the majority or all aircraft to early pushback times already drives the average departure throughput to high values even without any planning involved, not leaving much “room” for the planner to actually improve departure throughput. Furthermore, late concentration of pushback times in the schedule, without the presence of any large pushback activity gaps, drives the throughput very low whether planning is performed or not (“Bunch Back All” schedule).

On the other hand, the existence of a large pushback gap at the front end of the “Bunch Back Partial” schedule creates the necessary “room” for the planner’s throughput optimizing function to improve throughput performance by planning pushback operations earlier to ensure the necessary departure runway pressure. Consequently, in the case of this schedule, the planner contributes to the system the highest throughput improvement of about 2 departures / hour among all other schedule cases.

The departure delay and elapsed time results in Figure 6.32 and Figure 6.33 provide similar insight. However, compared to the departure throughput results, there appear to be more cases where the presence of the planner makes a favorable difference with respect to delay and elapsed time performance. Among all six schedules that were tested, only the original schedule did not

record any delay or elapsed time benefits when planning was applied to the departure process³⁹.

As it was discussed in the re-planning robustness analysis in section 6.4.3, the planner tends to prioritize the improvement of throughput performance, but when throughput issues are already addressed and departure throughput is already at a high level, the planner is in most cases able to introduce to the system benefits in the form of improved departure delay and elapsed time performance. Such an observation is more evident in the cases where the planner was not able to offer much in the improvement of departure throughput. These are the cases of the “Bunch Front All”, “Bunch Front Partial” and “Bunch Back All” schedules, for which nevertheless, there is a wide range of delay and elapsed time improvements recorded, with savings that in some cases reach up to the level of 1 to 1.3 min / aircraft (e.g. “Bunch Back All” and “Bunch Back Partial” schedules).

Examining together the throughput trends in Figure 6.31 and the delay and elapsed time trends in the pair of Figure 6.32 and Figure 6.33, we can draw some conclusions that should be further researched. It seems that, if runway and gate operations planning is available, scheduling pushback operations close to each other and as early as possible is a good way to ensure that departure runway pressure and therefore departure throughput are maintained at a high level, while at the same time, cumulative delays and elapsed times are prevented from raising uncontrollably.

³⁹ This was also seen in the benefits analysis in section 6.3.

If planning is not available, the previous method can lead to excessively high system delays. Therefore, in such a case, the next best option seems to be the regularization (as much as possible) of the pushback schedule. As shown in Figure 6.31, this seems to be a secure way to maintain a high departure throughput, comparable to the throughput achieved with closely spaced early pushback operations. At the same time, a regular schedule is not likely to yield departure delays and elapsed times that are very much higher than those of any other schedule that was tested (Figure 6.32 and Figure 6.33).

6.7 The Effect of Demand Weight Class Mix

The first stage of the decomposed two-stage runway operations planning algorithm uses the weight class mix (or else referred to as “fleet mix”) as its optimization basis (“decomposition pivot element” introduced and defined in section 3.4.3). Therefore, the weight class mix is an important factor in the generation of runway operations plans and its effect on operational benefits achieved through the use of the algorithm is investigated in this section.

The weight class mix of all pushback schedules of 15 aircraft (regular, i.e. with evenly-spaced pushback times or non-regular) used so far, is the one of the “nominal” schedule that was introduced earlier: Two (2) Heavies (13.33 %), ten (10) Large (66.67 %) and three (3) Small (20 %) aircraft, which is representative of the traffic mix encountered at many US airports. As it is explained in section D.1 of Appendix D, the presence of two Heavy aircraft in this nominal schedule and their location within the weight class sequence used for generating a runway

operations plan can impact the throughput resulting from the planner and consequently can affect the throughput and delay benefits that can possibly emanate from the use of the two-stage algorithm. Such issues are examined here by testing the behavior of the planning algorithm when the latter is acting on two original pushback schedules with two new weight class mix compositions, in addition to the one that was used so far in all simulation test cases:

- The first schedule consists of only one (out of 15) Heavy aircraft and 14 Large aircraft, i.e. 6.67 % Heavies, 93.33 % Large and 0 % Small and will be referred to as the “nearly homogeneous” schedule and
- The second schedule has a weight class mix composition of 33.33 % of each of the three possible weight classes, i.e. 5 Heavies (out of 15), 5 Large and 5 Small aircraft and will be referred to as the “balanced” schedule.

To isolate the study on the weight class mix characteristics of the pushback schedule, consecutive pushback times are evenly spaced at 60 sec apart from each other.

6.7.1 Scenarios WITHOUT Planning

Without yet considering the existence of a departure planning function, the results indicate that the weight class mix of the original pushback schedule can affect the level of departure throughput that can be expected on average (Figure 6.34).

As it was explained in Chapter 3, the weight class mix of the input schedule and the takeoff

sequencing of weight classes are strongly coupled to the departure throughput. That is because, based on the wake vortex separation criteria between successive departures on the same runway (Table C.2), the weight class sequencing at the runway determines the total time necessary for a group of operations to be completely serviced on the departure runway.

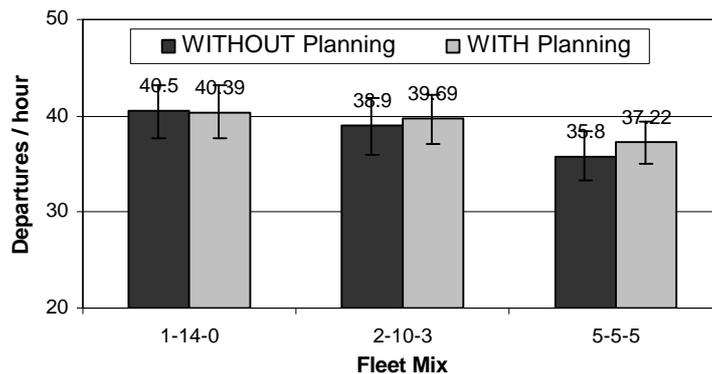


Figure 6.34: Effect of Schedule Weight Class Mix on Departure Throughput

It is obvious that the “nearly homogeneous” weight class mix leads to a much smaller number of possible weight class sequences (compared to the nominal and the balanced schedules) and consequently, it offers a much smaller range of possible throughput outcomes. This means that random takeoff position swaps that can affect the final departure throughput are much less likely to occur within a nearly homogeneous schedule. That is why the unplanned departure throughput is on average higher for the nearly homogeneous schedule than it is for the nominal or the balanced schedule, which have a much larger number of possible throughput outcomes. However, there is no such large disparity between the average delays and elapsed times for the three schedules with the different weight class mix compositions.

As observed in Figure 6.35 and Figure 6.36, cumulative departure delays appear to be on average the same for the three different weight class mix compositions, but the balanced schedule leads to higher cumulative elapsed times. That is because such a pushback schedule with a balanced fleet mix composition includes more heavy aircraft than any of the other two schedules, which naturally take longer to taxi out to the runway.

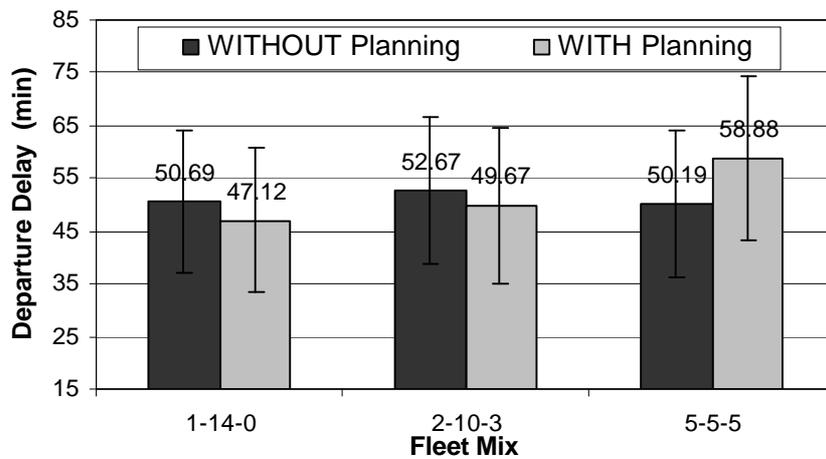


Figure 6.35: Effect of Schedule Weight Class Mix on Departure Delay

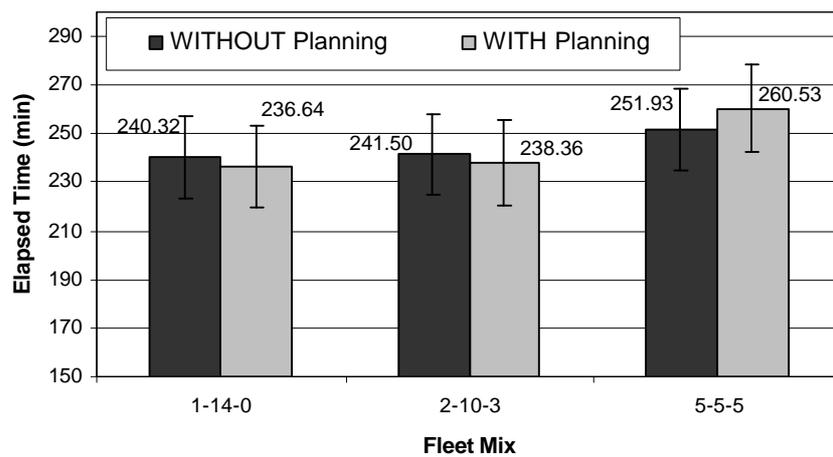


Figure 6.36: Effect of Schedule Weight Class Mix on Elapsed Time

In the presence of a departure planning function, the next section will examine how the above issues reflect to the ability of the planner to improve the airport's operational performance.

6.7.2 Scenarios WITH Planning

In the case of the balanced schedule, the larger range of possible weight class sequences that can be generated provides the planner's 1st stage throughput maximization function with more throughput options to test in order to plan a weight class sequence of takeoffs that maximizes departure throughput. That is why in Figure 6.34, the balanced schedule case yields the highest departure throughput improvement when planning is performed, even though the throughput values achieved (both with and without planning) are lower than those achieved with the nearly homogeneous or the nominal schedule. However, the highest throughput improvement comes to the price of having every aircraft absorb additional departure delay and elapsed time compared to the unplanned case.

With the help of the planner, it seems that the nominal schedule performs quite well both in terms of throughput and delay / elapsed time performance benefits. However, this nominal schedule has a weight class mix that could easily be altered by uncertainty in a real-world airport operational environment and therefore, such a fleet mix composition in the pushback schedule cannot be easy to control and maintain as desired.

Based on the limited results in Figure 6.34, Figure 6.35 and Figure 6.36, it seems reasonable to suggest that, if at any given operational period, it is necessary to improve and maintain at a high

level the performance of departures with respect to all primary performance characteristics of the runway and taxiway system, i.e. departure throughput and delays / elapsed times, the safest way to achieve that, without the need for a departure planning system, is to temporarily enforce “homogenization” of the fleet mix composition of the original pushback schedule that is given to the controllers.

6.7.3 Stability Analysis

The larger number of possible weight class sequences emanating from a schedule with a balanced fleet mix composition can explain the higher instability with respect to aircraft class sequence that is observed in Figure 6.37 below, compared to the stability behavior observed in Figure 6.27 for the nominal schedule.

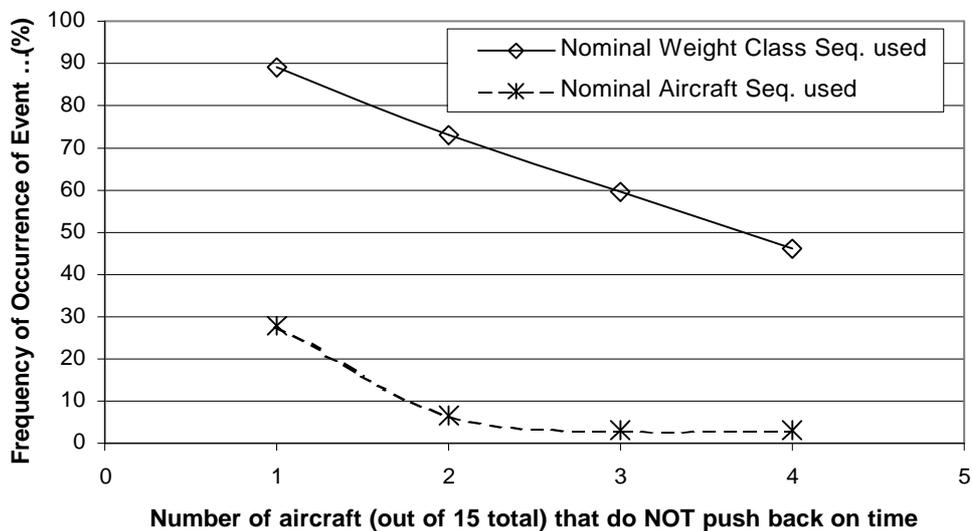


Figure 6.37: Stability of Algorithm Solutions - Pushback Uncertainty Standard Deviation of 120 sec, Balanced Weight Class Mix (Heavy, Large and Small, 33.33 % each) in the Original Pushback Schedule

Stability with respect to the weight class sequence is high. In the face of pushback schedule uncertainty, the plurality of weight classes in the mix allows for the majority of necessary changes in aircraft takeoff positions to occur within the same weight class subgroup without needing to change as often the weight class sequence that is used for the generated plan solution. Nevertheless, that very same weight class plurality significantly reduces the stability of the algorithm's output with respect to the aircraft sequence, exactly because it makes it much easier for aircraft to have their takeoff sequence position changed as a response to the impact of pushback uncertainty on the original pushback schedule.

6.8 Pushback Position Shifts vs. Takeoff Position Shifts

Given the stability and robustness behavior of the algorithm output as it was presented in previous sections, the values of the following two metrics were also examined, in order to investigate their relation to the observed behavior of the algorithm output:

1. The average number of pushback position shifts (aircraft swaps) relative to the nominal case, which represents the average level of change introduced in the expected pushback schedule input due to uncertainty and
2. The average number of takeoff position shifts relative to the nominal case, which represents the average level of change in the resulting runway operations plan due to the pushback input changes.

The above two metrics are calculated by averaging position shifts over all 500 runs of each simulation case that was run under the operational scenario WITH ADAPTIVE Planning. They

are presented in Figure 6.38 and Figure 6.39 as a function of low numbers (0 to 4) of aircraft in the schedule that do not push back on time. Even though these results are for a pushback delay standard deviation of 120 sec, results indicate that they are representative of the behavior of pushback and takeoff shifts for lower and higher standard deviation values.

The results in Figure 6.38 were derived by averaging position shifts over all runs of each simulation case, regardless of whether the nominal weight class and aircraft sequence were used or not to produce the final algorithm solution. Even though the average number of position shifts introduced in the pushback schedule does not change significantly as a function of the number of aircraft in the schedule that do not push back on time, nevertheless, it appears that the average number of takeoff position shifts rises significantly as more aircraft have their pushback times affected by uncertainty.

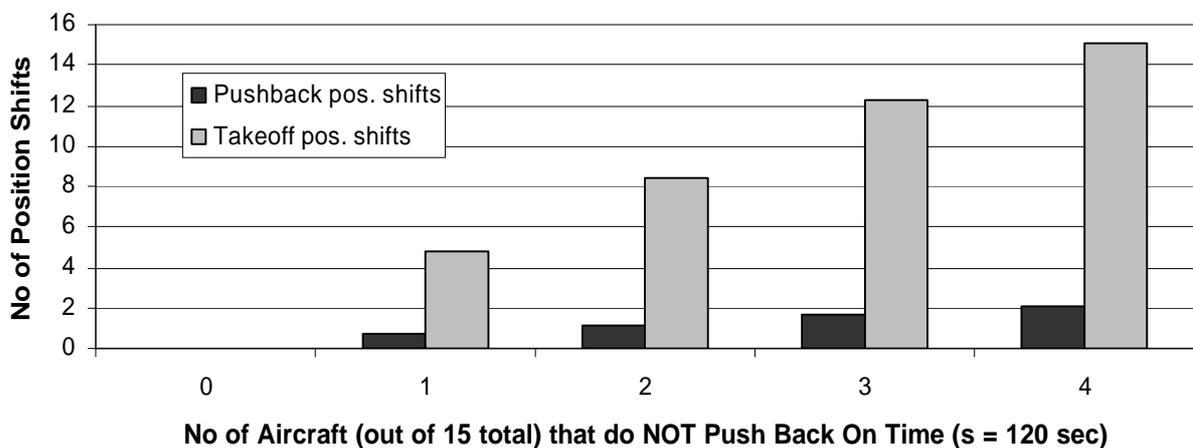


Figure 6.38: Change in Algorithm Input vs. Algorithm Output (All 500 Runs)

In the stability analysis of section 6.5, it was shown that, as the number of aircraft that do not push back on time increases, it is less likely that any given simulation run will produce a solution in which the optimal weight class sequence of the nominal case is used (Figure 6.27). So, the higher the number of aircraft that do not push back on time, the lower the percentage of runs (among all 500 runs of each simulation case) that is expected to lead to zero takeoff position shifts and therefore, the higher the overall takeoff position shift average value rises.

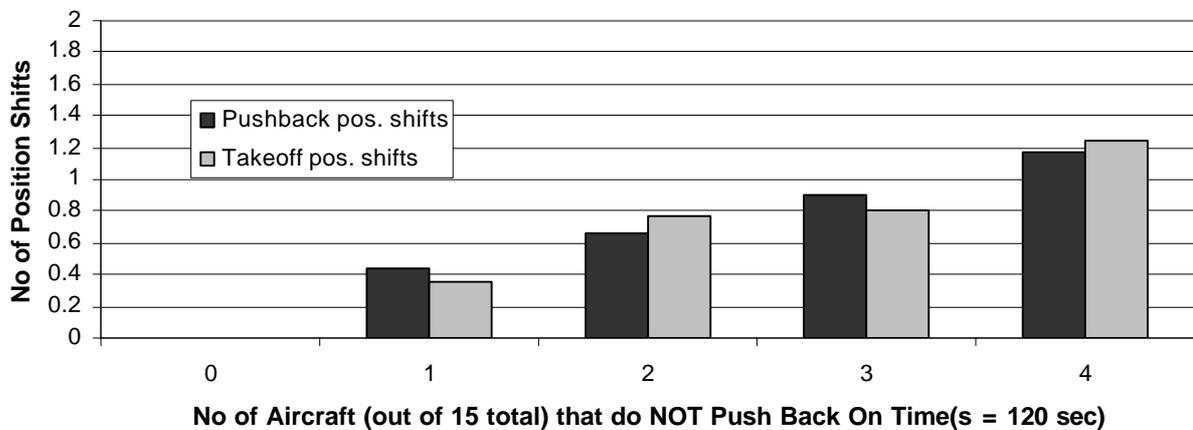


Figure 6.39: Change in Algorithm Input vs. Algorithm Output (Only Runs with Optimal Class Sequence)

The results in Figure 6.39 were derived by averaging position shifts over only those runs (among the 500 runs of each simulation case) in which the nominal weight class sequence was used. For the reasons described above, there is a rise in average takeoff position shifts as a function of the number of aircraft in the input pushback schedule that do not push back on time, but this time, as expected, the rise is not as steep as the one in Figure 6.38 and the average value pushback and takeoff positions shifts are significantly lower.

6.9 Robustness and Stability In the Presence of Air Traffic Control Restrictions

To this point, the robustness and stability analysis of the algorithm output behavior was limited to scenarios with no active air traffic control restrictions. However, these limited scenarios can also be considered as being representative of the output behavior expected even when flow restrictions are active. That is because when air traffic flow restrictions, such as Miles In Trail or Ground Stops are introduced in the system in the realm of everyday real-world operations, in most cases they mandate changes in the pushback times of certain aircraft that are affected by the restrictions. Hence, it can be argued that the effect of flow restrictions on aircraft pushback times is similar to the “time-shifting” effect that operational uncertainty has on pushback times, which was actually modeled and tested in the previous sections in the absence of flow restrictions.

Chapter 7

CONCLUSIONS & IMPLICATIONS

Designing an algorithm for addressing the runway operations planning problem resulted to a set of research contributions in the field of airport surface operations management and the application of the algorithm in the example cases presented in Chapter 6 provided insight on airport surface operations in general and runway operations in particular.

Research contributions are presented at the beginning of this chapter and then, the main research conclusions are summarized, regarding the potential utility of the two-stage Runway Operations Planning algorithm. Future implications are also discussed, regarding how this research can contribute to surface operations planning and how the decomposition methodology⁴⁰, as it was defined at the beginning of Chapter 3, can be used in addressing other optimization problems with a structure similar to the one of the Runway Operations Planning problem.

⁴⁰ ... based on which the two-stage algorithm was constructed ...

7.1 Summary of Research Contributions

At the beginning of this research effort, real-world airport operations were observed at Boston's Logan International Airport [70], [72], [73] in order to identify the main issues associated with the airport departure flow, such as the various departure flow constraints that introduce delays and inefficiencies and contribute to the low predictability of departures [74]. Based on these field observations, a conceptual architecture was proposed ([1] and Appendix B) as a structured way to address the problem of planning airport ground operations. In addition, the runway was identified to be a key departure flow constraint and thus, research work focused primarily on runway operations planning. The following three sections outline the contributions of this work in terms of:

- The decomposition methodology used to analyze and simplify the optimization problem in hand,
- The two-stage decomposed algorithm that was generated as an application of the decomposition methodology to the Runway Operations Planning problem and
- The details of using the runway operations plans suggested by the above algorithm.

7.1.1 Decomposing the Runway Operations Planning Problem

A decomposition-based methodology for simplifying complex optimization problems of a particular structure was abstractly introduced in Chapter 3. The methodology was then used to design an algorithm for solving the Runway Operations Planning problem.

The *decomposition-based methodology* represents an extension of common design theory decomposition techniques to the design of optimization algorithms and can potentially be applied to a general class of optimization problems with various objectives and a structure in which:

- A particular problem parameter or a group of parameters is more “dominant” on the problem performance functions than other relevant parameters,
- Different problem parameters take values under very different time scales, i.e. some change value faster than others and
- Some pieces of information are available earlier and with more certainty than others.

Such characteristics can assist in parsing the sets of system functions and system parameters in loosely coupled or non-interacting subsets, which allows the “function-parameter mapping” to be treated independently in different stages of the solution methodology.

The application of decomposition concepts to the development of an algorithm for the Runway Operations Planning problem was based on the premise that algorithm design can follow the techniques and guidelines that are common in design theory, such as Functional and Relational Decomposition, as well as Hierarchical Decomposition concepts⁴¹. Based on the latter, a “*dynamic decomposition*” component was deployed in the form of a temporal (time scale) analysis of problem properties, which takes advantage of the different time scales of the various system attributes in order to examine their dynamic behavior, as well as the degree of their impact on system performance. This component is similar to the “frequency separation” introduced by Gershwin [54]. However, when dealing with any dynamic quantity, the central

⁴¹ ... which have largely been studied in the context of multiple time-scale problems, e.g. in control theory ...

assumption behind Gershwin's proposition is to treat quantities that vary much more slowly as static and model quantities that vary much faster in a way that ignores the details of their variations. An assumption such as the above is not necessary in the dynamic decomposition as it is used in this thesis, i.e. in the form of a temporal (time scale) analysis of the problem properties with an objective to assist parsing of the system functions and parameters in subsets.

In decomposing the Runway Operations Planning problem, the concept of the "*decomposition pivot element*" was introduced and its selection among the problem attributes was discussed. The "pivot element" was defined as the "non-unique (non-fixed) problem parameter around which the problem decomposition revolves". It is the problem parameter that is chosen as the basis for decomposing the problem and its solution approach into several stages. The first of these stages prioritizes optimization of the problem functions that are primarily coupled to and affected by the pivot element⁴².

The analysis of the "*time scale*", "*information certainty*" characteristics of problem parameters as described above can assist in the selection of the "pivot-element", by uncovering the problem parameters that are characterized by low volatility and therefore high reliability to plan based on them. In addition, since many Air Traffic Management (ATM) scenarios involve certain factors that are more impactful over the whole scenario than any other factor in the scenario, analysis of the "*degree of impact*" of problem parameters can also point to an appropriate "pivot-element". One way to determine the "degree of impact" of a factor would be to measure the change in the

⁴² This process is abstractly described in section 3.4.3 and outlined more specifically for the Runway Operations Planning problem in section 3.5.1.

magnitude of a system performance function that results from small perturbations of that particular factor. To understand this, we consider the problem of planning and scheduling airport ground operations, where runway throughput is one of the key performance functions. While there are several factors and parameters that can have an impact on this function, some have a more significant impact than others. Let us assume that a ground operations plan has been generated and that this plan has a specific level of runway throughput. Introducing a small change in the plan may or may not change the resulting throughput. If a specific flight in the plan is switched with another flight (different aircraft tail number) of the same weight class, then throughput is not affected, but if the new flight is operated by an aircraft of a different weight class than the one it is substituting, then runway throughput may change due to different wake vortex separation requirements between the newly-introduced aircraft and its trailing aircraft. This shows that the factor “aircraft weight class” is more dominant than the factor “aircraft tail number”, but does not mean that “aircraft weight class” will always be the dominant factor.

As problem conditions may change with time, the pivot element and the form and characteristics of the resulting decomposition might change also. This was demonstrated in Chapter 3 through an example which showed that, depending on the severity of active air traffic flow restrictions, such as Miles In Trail (MIT) separation or Expected Departure Clearance Time (EDCT) time window requirements, the factor “downstream flow restrictions” may assume a more dominant role in the problem of Runway Operations Planning than that of the “aircraft weight class”, even though flow restrictions typically do not affect every single flight involved in the planning process, while, on the other hand, every flight has a weight class associated with it. Selection of the pivot element relies on studying the problem’s functional requirements (Functional

Decomposition) and analyzing the coupling behavior between those functional requirements and the problem parameters (Relational Decomposition).

7.1.2 Runway Operations Planning (ROP) Algorithm

The application of decomposition to the specific problem of *Runway Operations Planning (ROP)* represents a contribution to the field of Surface Operations Management (SOM), since Runway Operations Planning is an important subtask of SOM and the success of ROP largely determines the efficiency of the airport system. As a result of the decomposition-based algorithm design methodology⁴³, a “two stage” optimization algorithm was developed for solving the Runway Operations Planning problem to determine the optimal departure schedule. An overview of the algorithm was given in Figure 3.2. The two-stage algorithm is robust to unforeseen events that can impact pushback operations (section 6.4), which potentially provides the flexibility of adjusting runway operations plans at a tactical level in order to cope with the highly dynamic events that occur within an airport environment.

Based on specific (often competing) objectives and subject to a number of often very strict constraints, the solution generated by the proposed algorithm has the form of a set of aircraft departure sequences and time schedules. A planning solution can be generated quite fast by performing both stages of the algorithm but also the second stage can be solved independently as an emergency response to last-minute unforeseen events. The planner has the opportunity to solve in the first stage a subset of the ROP problem with respect to those problem parameters

⁴³ A one-stage approach to planning runway operations is discussed in Appendix G.

that may be “dominant” at the moment, without ignoring the rest of the parameters, but just deferring them to the remaining stage(s) of the process. The results from the first stage of the planning algorithm are to be used by all controllers (upstream and downstream) as a basis for their decisions. This provides a potential planning link between upstream and downstream controller positions, however, still allowing flexibility to the downstream (Local) controllers to exercise control without being as much affected by the decisions of upstream (Gate) controllers.

In the process of designing the two-stage Runway Operations Planning algorithm, it was realized that crossing of taxiing aircraft imposes one of the major constraints on runway operations. For that reason, the proposed decomposed approach for solving the Runway Operations Planning problem also included *interaction between arriving and departing traffic* through the planning of runway time dedicated to runway crossings, which is an issue that was overlooked in other research approaches to airport ground operations problems. No previous optimization approach had addressed the Runway Operations Planning problem to the detailed level of scheduling operations on an “aircraft-to-aircraft” basis, including all types of operations (departures, arrivals and aircraft crossings) that request runway time, while at the same time taking into account uncertainty in pushback and taxi operations.

7.1.3 Using the runway operations plans suggested by the ROP algorithm

The usefulness and applicability of the results of the two-stage Runway Operations Planning algorithm was investigated through the testing methodology described in Figure 6.5. In the process, it was realized that, optimization of the sequence and times of runway operations

(departures, arrivals and crossings) at an active runway cannot be isolated from the rest of the taxi path that departing aircraft follow from their gates until they are airborne. This is due to the fact that airport surface operations, whether they take place at a gate, on the taxiway system or at a runway, are inherently characterized by high unpredictability emanating from a multitude of unforeseen events that can possibly affect aircraft movement on the airport surface. In addition, uncertainty in the way that human operators of aircraft (pilots) and human controllers of traffic (tower air traffic controllers) respond to different events adds to the unpredictability of airport surface operations.

For the above reasons, when an optimization approach was suggested for solving the Runway Operations Planning problem, *the solutions from the approach became useful only after they were “translated” to gate pushback suggestions*. A queuing adjustment methodology was also presented, which claimed that runway queuing is not necessarily an inevitable manifestation of undesired airport surface congestion but instead it is actually *important in maintaining pressure on the runway and thus ensuring runway operations close to or at throughput optimality*. Therefore, in order to take into account the uncertainty in airport operations, *airport surface queuing must be considered in the process of translating runway plans to pushback plans*.

It was determined that, when calculating a planned pushback time from a planned take off time for each aircraft, in addition to subtracting the average unimpeded taxi time, an amount of queuing time adjustment corresponding to the expected queuing also needs to be subtracted in order to generate a realistic Pushback Plan. If the expected queuing is assumed very high, then aircraft tend to be planned to push back earlier, which can exercise pressure on the runway and

lead to increased throughput, but it can also lead to excessive taxi times and queuing delay periods for each individual aircraft. If queuing is assumed low, then the risk of excessive delays is avoided but then, the lack of the appropriate level of runway pressure can lead to runway idle time periods and reduced throughput performance even during peak traffic periods.

In Chapter 4, *a methodology was developed for determining just the appropriate amount of queuing* that can result to a plan of pushback operations that occur early enough to ensure sufficient pressure on the departure runway, while at the same time prevents surface delays from “skyrocketing”. An analysis of taxi time data from Boston’s Logan airport was performed to determine the level of expected queuing to be used in the planning process, i.e. the level of queuing that balances the tradeoff between taxi delays and runway throughput. The amount of appropriate queuing time adjustment was found to have a quadratic dependence on the estimated queue size that each aircraft is expected to encounter ahead of it. This means that a different value of queuing time adjustment was used for each aircraft being planned, as the number of aircraft expected to be ahead of it changed.

7.2 Summary of Research Findings

The operational performance, robustness and stability of the Runway Operations Planning algorithm’s output were tested based on a benchmark airport (Figure 5.1) that closely corresponds to Boston’s Logan runway configuration 22L / 22R / 27. The group of departing flights involved in the tests was a set of fifteen (15) aircraft with scheduled pushback times that

span within a period of about 14 minutes. We tested the effect of several problem parameters on:

- a) The ability of the two-stage runway operations planning algorithm to improve the operational performance of the departure flow and
- b) The robustness and stability of the algorithm output which, in a real-world airport tower environment, is to be given to air traffic controllers for implementation.

Our main conclusions are reported in the following paragraphs.

7.2.1 Effect of Active Flow Restrictions

In the absence of flow restrictions, it was found that the introduction of planning could yield departure throughput improvement (Table 6.2). This however, came to the expense of small additional delay absorbed by each aircraft, because the planning algorithm prioritized on improving the departure throughput performance of the system. Furthermore, because field observations at Boston's Logan Airport ([72], [73]) revealed that air traffic controllers and air traffic management coordinators give significant importance on active flow restrictions, the operational performance of the two-stage algorithm was also tested under a traffic scenario with active Miles In Trail restrictions.

Before any planning was performed, flow restrictions introduced inefficiencies in the departure flow that were even higher than those in the unrestricted case and deteriorate airport performance by reducing the departure throughput and introducing higher delays in the system. However, as expected, the optimizing effect of the two-stage algorithm leads to higher benefits than the ones introduced in the unrestricted traffic scenario.

7.2.2 Effect of Pushback Timing

When traffic demand is high, it is unlikely that gaps will appear in the pushback schedule, which ensures constant departure pressure. However, the effect of irregular pushback timing, i.e. timing that significantly deviates from evenly spaced pushback times, on the operational utility of the two-stage algorithm was also tested.

The level of regularity of pushback times can have an impact on the airport performance level before any planning is attempted. Typically, “bunching” of pushback operations ensures departure pressure and a high throughput, unless a pushback gap is introduced in the schedule. One or more pushback gaps can undermine departure throughput by reducing the pressure on the departure runway, while at the same time, they can increase taxi out times and runway queuing delays. Given this, the level of regularity of pushback times and the location of possible inactivity gaps in the pushback schedule can affect the “room” left for improving departure operations through planning. The existence of an inactivity gap can reduce the pressure exercised on the runway by frequent pushback operations and therefore, it can reduce departure throughput and open up some opportunities for the departure planning function to improve operational performance. On the other hand, a very dense pushback schedule with high departure throughput and no inactivity gaps does not leave much “room” for throughput improvement by using the two-stage algorithm. However, there are cases that, together with a high throughput, high delays manifest before planning is introduced and therefore, there is “room” for the planning algorithm to improve the time-based performance of the runway system.

The regularization of the pushback schedule was found to be a good “balance” point in the

throughput / delay tradeoff, with or without the presence of planning. A regular pushback schedule seems to maintain a high departure throughput, comparable to the throughput achieved with closely spaced early pushback operations (Figure 6.31). Also, a regular schedule is likely to yield departure delays and elapsed times that are lower or at most slightly higher than those of any other schedule that was tested (Figure 6.32 and Figure 6.33).

7.2.3 Effect of the Weight Class Mix

Frequently there are periods of time when the departure demand is characterized by only one or two weight classes in the traffic mix profile, as opposed to all three (Heavy, Large and Small) that were present in most simulation tests in Chapter 6. It was realized that, the weight class characteristics of the aircraft involved in planning directly affect the usefulness of performing runway operations planning because the level of weight class diversity or homogeneity (lack of diversity) of the pushback schedule affects the size of the range of possible throughput outcomes from all possible weight class sequence permutations for the given set of aircraft.

For example, in the case of an entirely homogeneous pushback schedule, a two-stage optimization approach with the “weight class” as the decomposition pivot element may not be useful, because there is only one possible weight class sequence and that does not offer any room for throughput maximization. Even for a less homogeneous pushback schedule with one aircraft having a different weight class than the remaining involved aircraft, e.g. a Heavy aircraft among all other Large aircraft, the maximum throughput solution is apparent and a two-stage algorithm may still not be necessary. Therefore, a two-stage decomposed runway operations planning

algorithm may not be appropriate for pushback schedules with all possible weight class mix compositions. For that reason, in addition to the original “nominal” pushback schedule, simulation tests were also performed for two other schedules: a) a nearly homogeneous one with only one Heavy aircraft among a total of 15 and b) a schedule with a totally balanced fleet mix composition.

Without any planning, it was realized that the departure throughput is on average higher for the nearly homogeneous schedule than it is for the nominal or the balanced schedule, because the latter have a much larger number of possible throughput outcomes and the former is much less vulnerable to random takeoff position swaps that can affect the final departure throughput. Also, cumulative departure delays appeared to be on average the same for the three different weight class mix compositions, with the balanced schedule leading to higher cumulative elapsed times because it includes more slow-taxiing heavy aircraft than any of the other two schedules.

When planning is performed, a balanced schedule was found to lead to the highest departure throughput improvement, even though the throughput values achieved (both with and without planning) are lower than those achieved with the homogeneous or the nominal schedule. This highest throughput improvement came to the price of having every aircraft absorb additional departure delay and elapsed time compared to the unplanned case.

Given the above, if a “preferred” fleet mix with respect to operational performance had to be defined, it would seem reasonable to suggest “homogenization” of the fleet mix composition of

the original pushback schedule, in order to improve and maintain a high level of departure performance with respect to departure throughput and delays / elapsed times, without the need to deploy the two-stage runway operations planning algorithm.

7.2.4 Effect of Pushback Uncertainty

In the face of unexpected pushback events that change the exact timing of pushback operations⁴⁴, the level of robustness and stability of solutions generated by the planning algorithm determines how beneficial the algorithm can be in improving the airport's operational performance. Uncertainty perturbations were introduced in the pushback process and plan robustness was measured with respect to operational performance indicators such as the departure throughput and departure delays resulting from the runway operations plans suggested by the algorithm. Also, the stability metrics used were based on the weight class and aircraft sequences suggested by the two-stage algorithm.

The stability behavior of the algorithm output was found to depend strongly on the number of aircraft that have their pushback times affected by uncertainty. The algorithm output remains quite stable only for a low number of aircraft that do not push back on time but manifests the same stability behavior regardless of how high or low the standard deviation of the pushback uncertainty is. Therefore, if the number of aircraft that do not push back on time is somehow kept constant, the two-stage algorithm can be expected to not lose or gain any stability performance as various local or remote events change the level of uncertainty existing in airport

⁴⁴ ... which is what the algorithm used as an input.

operations.

It was found that, despite the impact of uncertainty, planning seems to lead to a fairly robust operational performance across a wide range of uncertainty levels (Figure 6.23 and Figure 6.24), i.e. average departure throughput is not compromised and average departure delays do not increase. So, when a new pushback plan cannot be generated to replace a previous plan that was affected by pushback uncertainty, the resulting throughput and delays should be expected to remain on average unaffected. Also, as demonstrated in Figure 6.25 and Figure 6.26 for a wide range of uncertainty parameter values, the throughput and delay benefits experienced in the absence of uncertainty (i.e., No of aircraft that do NOT push back on time = 0) are not compromised by pushback uncertainty regardless of the severity of its impact.

Summarizing robustness and stability conclusions, when uncertainty affects pushback times:

- As uncertainty becomes more “impactful”, the algorithm output tends to be quite unstable in terms of the weight class and aircraft sequences suggested by the algorithm.
- The algorithm output demonstrates a robust behavior with respect to the average departure throughput and delay performance achieved by the generated runway and gate operations plans. Therefore, generating a runway and gate operations plan once (no adaptive planning performed) seems to be enough to ensure that at least throughput performance is maintained at a high level, since the planner prioritizes departure throughput maximization. It can be expected that a rising number of aircraft not pushing back on time will not have a significant effect on performance and the only issue of concern will be the stability of the generated

plans in terms of aircraft and weight class sequence.

- After planning is performed, there are moderate benefits with respect to departure delays that can be achieved in the face of pushback uncertainty, if adaptive planning is performed and a new runway operations plan is generated.

7.3 Implications for Real-World Problems

In an airport operations context, the main goals behind parsing the Runway Operations Planning problem in two stages and developing a decomposed algorithm to solve it were:

- a) To construct a new method for optimizing runway operations, which would provide some level of operational performance gain compared to the current methods of handling ground traffic
- b) To achieve flexibility in tactical planning and adjusting plans in a short time in order to cope with the highly dynamic events that occur within an airport environment and
- c) To plan runway operations in a way that is robust to change, e.g. unforeseen events on the airport surface.

Similar planning flexibility and robustness ideas can be developed for other types of systems with characteristics that resemble those of the ROP problem. In fact, this becomes easier to imagine if we abstract the runway as a service provider that has to provide several different types of service (i.e. departures, arrivals and crossings) to various types of customers (i.e. aircraft that

are Heavy, Large or Small, jet or prop, heavily restricted or not). A multiple – stage algorithm for tactical operations planning can potentially match pretty well the planning needs of any such system.

For example, tactical planning and sequencing of operations of bank tellers and customer service representatives at a bank branch with high daily traffic volume may benefit from a multiple-stage heuristic algorithm. Bank customers arrive at random times and each of them has a specific issue that needs to be addressed, whether it is as simple as withdrawing money or as complicated and time-consuming as opening a loan account. The expected demand for each specific type of service that the bank provides is quite volatile and changes at high frequency throughout the business day. Therefore, it is not reliable enough to be used by the bank as a basis for planning its service resources at a tactical level. However, the expected bank customer demand in terms of short vs. lengthy service requests, or in terms of demand for a teller operator vs. a desk representative is more or less expected to not vary significantly during one day. Of course, across a longer time horizon, such as a week or a month, the latter argument is not safe to make, because the expected demand may follow a highly variable trend, even within a week, depending on the way each bank under the same bank system (or any bank in each different bank system) performs its operations. All these demand characteristics for the particular example of “bank operations planning”, are most likely quite easy to verify with the appropriate collection of data at one or more bank branches. Application of a two-stage planning algorithm on a system like this one can help to:

- a) Initially plan the staffing needs of the bank branch at a department level and then
- b) Continue planning the staffing needs at as many subsequent levels as deemed necessary,

based on the branch's particular nature of operations, i.e. plan within each department (e.g. business vs. retail banking), then inside a specific department plan with respect to type of service performed among all the services that the department offers and so forth.

7.4 Future Research

The simulation methods that were implemented in this thesis in order to research the performance of the proposed two-stage decomposed algorithm, considered a significant amount of *controllable parameters*, but in the scope of this thesis particular values were chosen for most of them. However, there is enough room for future further testing of the algorithm's behavior for different values of some or all of these parameters, such as:

- *The total number of scheduled departing aircraft considered for planning*: Even though the algorithm and the optimization behind it can handle any small or large number of aircraft, the larger that number becomes, the longer it takes for the algorithm to generate a feasible optimal solution. In addition, given the dynamic nature of airport ground operations and the high uncertainty associated with aircraft movements, it usually has no practical value to plan runway operations for a large number of aircraft a long time before they actually push back from their gates and enter the movement areas of the airport surface.
- *The parameters that determine crossing constraints*, such as the maximum allowable crossing delay and the capacity of the taxiway segments between the two parallel runways under the runway configuration selected in Figure 5.1.
- *The number, type and magnitude of air traffic flow constraints* that are typically active at US

airports and were modeled in the simulation presented in the thesis.

Also, a particular runway configuration was chosen for testing, which is one of the most commonly used at Boston's Logan airport and other busy US airports. Nevertheless, it is possible to investigate how *other configurations and other airport surface geometries* could affect the operational behavior of a runway operations planning algorithm, such as the one proposed through this work. From field observations at different airports, it has been documented ([72], [73]) that airport surface management is an airport-dependent process. Therefore, extension of the algorithm testing to other airports could uncover how this method of planning runway operations is affected by elements and issues that do not usually (or at all) manifest at Logan airport, such as hub-spoke airline operations.

One more element of this work where there is room for further research is the method described in Chapter 6 that determines the optimum curve to be used for calculating expected aircraft queuing times which are then used in "translating" generated runway operations plans to gate pushback plans. So far, the *queuing time calculations* were based on historical data acquired from Logan airport and therefore, in a sense, the same Logan queuing behavior was reproduced for our simulation purposes. Remembering that the main issue was the effect that queuing at the runway can have on departure throughput and cumulative taxi delay values and the inevitable tradeoff between the two, it becomes important to plan runway operations while at the same time optimizing the runway queue length. For example, there is no value in maximizing departure throughput by planning pushback operations as early as feasible in order to exercise constant pressure at the departure runway, if at the same time the resulting cumulative taxi delays are

excessive due to uncontrolled runway queue lengths. On the other hand, there is no value either in planning pushback operations later rather than earlier with the hope of keeping taxi delays low, because in that case, pressure on the runway may not be enough to maintain constant use of the available runway time and therefore departure throughput may be reduced. Therefore, the expected queuing calculations implemented during runway operations planning, could be based on testing with Monte Carlo simulations different values for the maximum allowable queuing size control.

One of the most important lessons learned through this research was that runway departure operations cannot and should not be managed isolated from the remaining types of operations that take place at an airport. Arriving aircraft with uncertain final approach paths and touchdown times⁴⁵, runway crossings, unpredictable aircraft taxiing and unobservable gate pushback operations are all an integral part of the same set of processes that comprise airport ground operations. Furthermore, since any airport is part of the National Airspace System (NAS), airport surface operations optimization cannot be successful without integrating ground traffic management decisions with all events and decisions associated with other elements of the NAS, such as other airports and en route airspace. This leads to a need for further research investigation of the issue mentioned in section 3.5.1, that has to do with the selection of the appropriate decomposition pivot element depending on the current situation at an airport. Assuming that a decomposition pivot element other than weight class has been selected for testing, e.g. downstream flow restrictions, a new set of simulation tools is required in order to conduct once again benefits and robustness analysis and to produce simulation results to be

⁴⁵ Arrival stream uncertainty was ignored for the purposes of this work.

compared with those from the cases tested in this thesis, where weight class was used as the pivot element.

Finally, regardless of the pivot element used, it is also important to conduct research on the necessary interfaces for integrating any departure decision-aiding tool with the rest of the already available or new decision-support systems that exist or are being built for elements of the airport other than the runway (e.g. taxiways and gates), as well as for the National Airspace System.

References

1. ANAGNOSTAKIS, I., IDRIS, H. R., CLARKE, J-P., FERON, E., HANSMAN, R.J., ODONI, A.R., HALL, W.D., “*A Conceptual Design of a Departure Planner Decision-Aid*,” International Center for Air Transportation, MIT, AIAA Progress in Aeronautics and Astronautics: Air Transportation Systems Engineering, American Institute of Aeronautics and Astronautics, Vol. 193, Chapter 21, Washington, District of Columbia, also at the 3rd USA/Europe International Air Traffic Management R & D seminar, ATM-2000, proceedings, session E: Airport, Ground Operations, Arrival and Departure Management, Naples, Italy, 13-16 June 2000, sponsored by FAA & Eurocontrol, available at <http://atm-seminar-2000.eurocontrol.fr/> [cited Oct. 6, 2000]
2. ANAGNOSTAKIS, I., CLARKE, J-P., BÖHME, D., VÖLCKERS, U., “*Runway Operations Planning and Control: Sequencing and Scheduling*,” AIAA Journal of Aircraft, November-December 2001, Vol. 38, No.6, also presented at the 34th Hawaii International Conference on System Sciences, Maui, Hawaii, January 3rd – 6th, 2001
3. ANAGNOSTAKIS, I., CLARKE, J-P., “*Runway Operations Planning: A Two-Stage Heuristic Algorithm*,” presented at the AIAA Aircraft, technology, Integration and Operations Forum, Los Angeles, CA, October 1st – 3rd, 2002
4. ANAGNOSTAKIS, I., CLARKE, J-P., “*Runway Operations Planning: A Two-Stage Solution Methodology*,” 36th Hawaii International Conference on System Sciences (HICSS-36), Big Island, Hawaii, January 6-9, 2003
5. ANDERSON, J., et al, “*Manageable Object-Oriented Development: Abstraction, Decomposition and Modeling*,” Proceedings of Tri-Ada '91, San Jose, California, October 21-25, 1991
6. ANDERSSON, K., CARR, F., FERON, E., HALL, W., “*Analysis and Modeling of Ground Operations at Hub Airports*,” presented at the 3rd USA/Europe Air Traffic Management R&D Seminar, Naples, Italy, 13rd - 16th June 2000
7. ANDREATTA, G., ODONI, A.R., RICETTA, O. [1993], “*Models for the Ground-Holding Problem*”, chapter in Large-Scale Computation and Information Processing in Air Traffic

- Control, L. Bianco and A.R. Odoni, editors, Springer-Verlag, Berlin, 1993, pp. 125-168
8. ANDREATTA, G., BRUNETTA, L., GUASTALLA, G., [1997] “*Multi-Airport Ground Holding Problem: A Heuristic Approach based on Priority Rules,*” Modeling and Simulation in Air Traffic Management, editors, L. Bianco, P. Dell’ Olmo, A.R. Odoni, Springer Verlag, pp. 71-89
 9. ANDREATTA, G., BRUNETTA, L., ODONI, A.R., RIGHI, L., STAMATOPOULOS, M.A., ZOGRAFOS, G. [1998], “*A set of Approximate and Compatible Models for Airport Strategic Planning on Airside and Landside,*” presented at the 2nd USA/EUROPE AIR TRAFFIC MANAGEMENT R&D SEMINAR, Orlando, 1st - 4th December 1998
 10. ATKINS, S., HALL, W., “*A Case for Integrating the CTAS Traffic Management Advisor and the Surface Management System,*” AIAA Guidance, Navigation, and Control Conference, Denver, CO, Aug. 14-17, 2000
 11. ATKINS, S., BRINTON, C., “*Concept Description and Development Plan for the Surface Management System,*” Journal of Air Traffic Control, Vol. 44, No. 1, January–March, 2002
 12. ATKINS, S., WALTON, D., “*Prediction and Control of Departure Balancing at Dallas / Fort Worth Airport,*” American Control Conference 2002 proceedings, Anchorage, Alaska, May 8-10, 2002
 13. AZARM, S., LI, WEI-CHU, “*Optimal Design Using Decomposition Methods,*” Technical Report TR-86-60, Dept. of Mechanical Engineering and Systems Research, University of Maryland, October 1986
 14. BALL M., HOFFMAN R., *Collaborative Decision-Making in Air Traffic Management: A Preliminary Assessment*, prepared by the National Center of Excellence in Aviation Operations Research, August 1998
 15. BEASLEY, J.E., KRISHNAMOORTHY, M., SHARAIHA, Y.M., ABRAMSON, D., “*Scheduling aircraft landings - the static case,*” Transportation Science, vol. 34, 2000, pp180-197
 16. BEASLEY, J.E., SONANDER, J., HAVELOCK, P., “*Scheduling Aircraft Landings at London Heathrow using a Population Heuristic,*” Journal of the Operational Research Society, vol. 52 2001, pp483-493

17. BENNETT, W. H., BARAS, J. S., “*Decomposition and Decentralized Control System Design: A Review of Frequency Domain Methods*,” presented at the 24th IEEE Decision and Control Conference, 1985
18. BERTSIMAS, D., TSITSIKLIS, J. N., “*Introduction to Linear Optimization*,” Athena Scientific, Belmont, Massachusetts, 1997
19. BIANCO, L., DELL’OLMO, P., GIORDANI, S., “*Scheduling Models and Algorithms for TMA Traffic Management*,” Modeling and Simulation in Air Traffic Management, editors, L. Bianco, P. Dell’Olmo, A.R. Odoni, Springer Verlag, pp. 139-168, 1997
20. BITRAN, G., R., HAAS, E., A., HAX, A., C., “*Hierarchical Production Planning: A Single-Stage System*,” Operations research, Vol. 29, No. 4, July – August, 1981, pp. 717 - 743
21. BÖHME, D.: “*Airport Capacity Enhancement by Planning of Optimal Runway Occupancies*,” Proceeding of the 39th Internationales Wissenschaftliches Kolloquium der TU Ilmenau Volume 3, pp. 250-256, ISSN 0943-7207, Sept. 24 – 30, 1994
22. BÖHME, D., “*Improved Airport Surface Traffic Management by Planning, Problems, Concepts and a Solution – TARMAC*,” H. Winter (Ed.): Advanced Technologies for Air Traffic Flow Management. DLR – Seminar Series, Bonn, Germany, April 1994, in Lecture Notes in Control and Information Sciences, Springer Verlag, ISBN 3540198954
23. BOLANDER, M.A., SLATER, G.L., “*Analysis and Optimization of Departure Sequences*,” AIAA 2000-4475, Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA, Denver, CO, 14-17 August 2000
24. BOLANDER, M.A., “*Scheduling and Control Strategies for the Departure Problem in Air Traffic Control*,” Ph.D. Thesis, University of Cincinnati, 2000
25. BRINTON, C., KROZEL, J., CAPOZZI, B., ATKINS, S., “*Improved Taxi Prediction Algorithms for the Surface Management System*,” AIAA 2002-4857, Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA, Monterey, CA, 5-8 August 2002
26. BUSHNELL D.M., “*Civil Aeronautics: Problems, Solutions and Revolutions*,” Aerospace America, April 2003, pp. 28 – 32
27. CAROMICOLI, C. A., “*Time Scale Analysis Techniques for Flexible Manufacturing*

- Systems*,” MIT EECS Master of Science Thesis, MIT Laboratory for Information and Decision Systems Report LIDS-TH-1725, December 1987
28. CAROMICOLI, C. A., WILLSKY, A. S., GERSHWIN, S. B., “*Multiple Time Scale Analysis Techniques of Manufacturing Systems*,” MIT Laboratory for Information and Decision Systems Report LIDS-TH-1727, December 1987
 29. CAROMICOLI, C. A., WILLSKY, A. S., GERSHWIN, S. B., “*Multiple Time Scale Analysis of Manufacturing Systems*,” Analysis and Optimization of Systems, Proceedings of the 8th International Conference on Analysis and Optimization of Systems, Antibes, France, June 1988, 12 pages
 30. CARR, F., EVANS, A., CLARKE, J-P., FERON, E., “*QMAST: Queuing Model for Airport Surface Traffic*,” Massachusetts Institute of Technology, 2002, available at http://icat-server.mit.edu/~fcarr/online_model/ [cited April 2002]
 31. CARR, F., EVANS, A., CLARKE, J-P., FERON, E., “*Modeling and Control of Airport Queuing Dynamics under Severe Flow Restrictions*,” American Control Conference 2002 proceedings (ISBN 0-7803-7299-9), Anchorage, Alaska, May 8-10, 2002
 32. CARR, F., THEIS, G., FERON, E., CLARKE, J-P., “*Evaluation of Improved Pushback Forecasts Derived from Airline Ground Operations Data*,” Working paper, International Center for Air Transportation, Massachusetts Institute of Technology, July 2003
 33. CHANG, C., “*Flight Sequencing and Gate Assignment at Airport Hubs*,” Ph.D. Thesis, University of Maryland, College Park, 1994
 34. CHU, A. K., “*The Development and Application of the Manufacturing System Design Evaluation Tool and Performance Measurement Based on the Manufacturing System Design Decomposition*,” M.Sc. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, June 2000
 35. CLARKE, JOHN-PAUL, *unpublished research work*, International Center for Air Transportation, MIT, September 1999
 36. COOPER, W.W., MOHLEJI, S.C., BURKE, C.D., FOSTER, G., MILLS, M.J., “*Determination of Minimum Push-Back and Taxi-Out Time Predictability Needed For Near-Term Departure Scheduling using DEPARTS*” presented at the 4th USA/Europe Air Traffic Management

R&D Seminar, Santa Fe, New Mexico, USA, 3rd - 7th December 2001

37. CTAS publication web site, <http://www.ctas.arc.nasa.gov/publications/index.html>
38. DAREING, S.W., HOITOMT, D., "*Traffic Management and Airline Operations*," American Control Conference 2002 proceedings (ISBN 0-7803-7298-0), Anchorage, Alaska, May 8-10, 2002
39. DAVIS, W.J., "*A Generalized Decomposition Procedure and Its Application to Engineering Design*," ASME Journal of Mechanical Design, Vol. 100, 1978, pp.739 - 746
40. DEAR, R.G., "*The Dynamic Scheduling of Aircraft in the Near Terminal Area*," Ph.D. thesis, Flight Transportation Laboratory Report R76-9, M. I. T., Cambridge, MA, September 1976
41. DEAR, R.G., SHERIF, Y.S., "*An Algorithm for computer assisted sequencing and scheduling of terminal area operations*," Transportation Research A.25, 129-139, 1991
42. DEARMON, J.S., "*Predictive Models For Air Traffic Arrival Flows at Capacity-Constrained Airports*," Air Traffic Control Quarterly Journal, Volume 5, Number 4, 1997, pp. 265-279
43. DELCAIRE, B., FERON, E., "*Dealing with Airport Congestion: Development of Tactical Tools for the Departure Flows from a Large Airport*," SM Thesis, Technology and Policy Program, MIT, Cambridge, MA
44. DEPARTS research project summary at:
http://www.mitre.org/technology/mtp01/quads_charts/decision_support/cooper.shtml
45. DIPPE, D., "*A Planning System for Airport Surface Traffic Management*," ECAC APATSI and EC Workshop on SMGCS, Frankfurt, April 1994
46. DIPPE, D., "*The DLR Activities for Development, Test and Evaluation of a Ground Movement Management and Control System*," AIAA 95-3369-CP, Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA, Baltimore, MD, August 1995
47. DOBBS, D. C., "*Development of an Aerospace Manufacturing System Design Decomposition*," M.Sc. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, September 2000
48. DONOHUE, G.L., LASKA, W.D., "*UNITED STATES and EUROPEAN Airport Capacity*

- Assessment using the GMU Macroscopic Capacity Model (MCM),*” 3rd USA/Europe International Air Traffic Management R & D seminar, ATM-2000, proceedings, session E: Airport, Ground Operations, Arrival and Departure Management, Naples, Italy, 13-16 June 2000, sponsored by FAA & Eurocontrol, available at <http://atm-seminar-2000.eurocontrol.fr/> [cited Oct. 6, 2000]
49. DUDA, J. W., “*A Decomposition-Based Approach to Linking Strategy, Performance Measurement and Manufacturing System Design,*” PhD. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, September 2000
 50. FAA, BOS TWR 7110.11H, Standard Operating Procedures, U.S. DOT, FAA, Boston Control Tower
 51. FAA, ‘*Logan Airside Improvements Planning Project,*” Boston-Logan International Project, Boston, Massachusetts, 2000
 52. FAAH 7110.65L, Air Traffic Control manual, U.S. DOT.
 53. FERON, E. et al., “*The Departure Planner: A Conceptual Discussion,*” Massachusetts Institute of Technology, December 1997
 54. GERSHWIN, S. B., “*Manufacturing Systems Engineering,*” Prentice Hall, 1994
 55. GILBO, E.P., [1991], “*Airport Capacity Estimates for Various Weather Conditions,*” ATMS/ETMS Project Memorandum, Unisys Corporation and Volpe National Transportation Systems Center, Cambridge, MA
 56. GILBO, E.P., [1993], “*Airport Capacity: Representation, Estimation, Optimization,*” IEEE Transactions on Control Systems Technology, Vol. 1(3), pp. 144-154
 57. GLASS B.J., GAWDIAK, Y., “*Integrated Navigation Applications in Airport Surface Traffic Management,*” 4th International Conference on Integrated Navigation Systems, St. Petersburg, Russia, May 1997
 58. GLASS B.J., “*Automated Data Exchange and Fusion for Airport Surface Traffic Management,*” AIAA 97-3679, proceedings of the AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, Aug. 11-13, 1997, pp. 904 – 911
 59. GOTTELAND, J-B., DURAND, N., ALLIOT, J-M., PAGE, E., “*Aircraft Ground Traffic*

- Optimization*,” 4th USA/EUROPE AIR TRAFFIC MANAGEMENT R&D SEMINAR, Santa Fe, NM, USA December 3rd - 7th, 2001, also available at <http://www.recherche.enac.fr/opti/papers/notice.html?todo=abstract&type=articles&file=atm2001jb.txt> (ENAC literature database) and <http://atm2001.eurocontrol.fr/> (ATM 2001 conference website)
60. HALL, W.D., “*Efficient Capacity Allocation in a Collaborative Air Transportation System*,” Ph.D. Thesis, operations Research Center, MIT, Cambridge, MA, May 1999
 61. HALL, W.D., PETERSON, E. H., “*ASCENT: A Network-Wide Simulation of Air traffic Flow Management Incorporating Airline Decisions*,” Proceedings of the American Control Conference (ACC), Anchorage, AK, May 8-10, 2002
 62. HEBERT, J.E., DIETZ, D.C., “*Modeling and Analysis of an Airport Departure Process*,” Journal of Aircraft, Volume 34, No. 1, January-February 1997
 63. HESSELINK, H.H., VISSCHER, J.J.E., “*Design of Runway Planning and Sequencing*,” NLR-TR-97094L and in MANTEA/ISR-TEC-D5.1-057
 64. HESSELINK, H.H., PAUL, S., “*Planning Aircraft Movements in Airports with Constraint Satisfaction*,” Proceedings of the 3rd IMA Conference on Mathematics in Transport Planning and Control, Cardiff, Great-Britain, April 1-3, 1998, NLR-TP-98397, proceedings in ISBN 0-08-043430-4
 65. HESSELINK, H.H., BASJES, N., “*Mantea Departure Sequencer: Increasing Airport Capacity by Planning Optimal Sequences*,” 2nd USA/EUROPE AIR TRAFFIC MANAGEMENT R&D SEMINAR, Orlando, 1st- 4th December 1998
 66. HORANGIC, B.R., “*Some queuing models of airport delays*,” M.S. Thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1990
 67. HUCK, V., KAUPPINEN, S., “*Basic Level DMAN, Operational Concept of Use*,” Report of the Eurocontrol European Organization for the Safety of Air Navigation, September 2002
 68. <http://mscmga.ms.ic.ac.uk/jeb/epsr.html>
 69. http://www.massport.com/logan/pdf/c_stats_dec03.pdf: Massport’s monthly airport traffic summary – December 2003

70. ICAO (1995), *Aircraft Engine Emissions Data Bank*, Doc 9646-AN/943, 1st edition
71. IDRIS, H.R., DELCAIRE, B., ANAGNOSTAKIS, I., HALL, W.D., PUJET, N., FERON, E., HANSMAN, R.J., CLARKE, J-P., ODONI, A.R., “*Identification of Flow Constraint and Control Points in Departure Operations at Airport Systems*,” AIAA 98-4291, Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA, Boston, MA, August 1998, Vol.2, pp.947-956
72. IDRIS, H.R., DELCAIRE, B., ANAGNOSTAKIS, I., HALL, W.D., HANSMAN, R.J., FERON, E., CLARKE, J-P., ODONI, A.R., “*Observations of Departure Processes at Logan Airport to Support the Development of Departure Planning Tools*,” 2nd USA/EUROPE AIR TRAFFIC MANAGEMENT R&D SEMINAR, Orlando, 1st - 4th December 1998
73. IDRIS, H.R., ANAGNOSTAKIS, I., DELCAIRE, B., HANSMAN, R.J., CLARKE, J-P., FERON, E., ODONI, A.R., “*Observations of Departure Processes at Logan Airport to Support the Development of Departure Planning Tools*,” Air Traffic Control Quarterly Journal, Special Issue in Air Traffic Management, Volume 7, Number 4, 1999, pp. 229-257
74. IDRIS, H.R., “*Observation and Analysis of Departure Operations at Boston Logan International Airport*,” PhD. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, September 2000
75. IDRIS, H.R., CLARKE, J-P., BHUVA, R.R., KANG, L.S., “*Queuing Model for Taxi-Out Time Estimation*,” Air Traffic Control Quarterly Journal, Volume 10, Number 1, 2002
76. JOHNSON, C.W., ISAACSON, D.R., LEE, K.K., “*Expedite Departure Path (EDP) Operational Concept*,” MIT Lincoln Laboratory Report, No. NASA/A-3, August 1999
77. JOHNSON, R.C., BENSON, R.C., “*A Basic Two-Stage Decomposition Strategy for Design Optimization*,” 1984, ASME Transactions, Journal of Mechanisms, Transmissions and Automation in Design, Vol. 106, pp. 380 - 386
78. JOHNSON, R.C., BENSON, R.C., “*A Multi-Stage Decomposition Strategy for Design Optimization*,” 1984, ASME Transactions, Journal of Mechanisms, Transmissions and Automation in Design, Vol. 106, pp. 387 - 393
79. JUNG, Y., ISAACSON, D.R., “*Design Concept and Development Plan of the Expedite Departure Path (EDP)*,” AIAA-2002-5812, AIAA Aircraft, technology, Integration and

- Operations (ATIO) Forum, Los Angeles, CA, October 1st – 3rd, 2002
80. KANG, L.S., CLARKE, J-P., “*Degradable Airline Scheduling*,” Working paper, International Center for Air Transportation, Department of Aeronautics & Astronautics, Massachusetts Institute of Technology, January 2003
 81. KUEST, K. M., “*The Development of the Production System Design Decomposition Framework*,” M.Sc. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, May 1999
 82. KUSIAK, A., WANG, J., “*Decomposition of the Design Process*,” Journal of Mechanical Design, Transactions of the ASME, Vol. 115, No. 4, December 1993, pp. 687 - 695
 83. LAWSON, D. R., “*Surface Movement Advisor (SMA)*,” Journal of Air Traffic Control, January-March, 1998
 84. LEE, K. K., DAVIS, T. J., “*The Development of the Final Approach Spacing Tool (FAST): A Cooperative Controller-Engineer Design Approach*,” Journal of Control Engineering Practice, Vol. 4, No. 8, August, 1996, pp. 1161-1168
 85. LINCK, J., “*A Decomposition-Based Approach for Manufacturing System Design*,” PhD. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, May 2001
 86. MAGILL, S.A.N., TYLER, A.C.F., WILKINSON, E.T., FINCH, W., “*Computer-Based Tools for Assisting Air Traffic Controllers with Arrivals Flow Management*,” Royal Signals and Radar Establishment Report No. 88001, September 1988
 87. MAGILL, S.A.N., “*Departure Manager Feasibility Report*,” Report (Released Issue) of the Eurocontrol European Organization for the Safety of Air Navigation, November 1999, available at <http://www.eurocontrol.int/odt/documents/library.html>
 88. MAIER, M.W., RECHTIN, E., “*The Art of Systems Architecting*,” CRC Press, Boca Raton, FL, 2000
 89. MATLOG: LOGISTICS ENGINEERING MATLAB TOOLBOX, Department of Industrial Engineering, North Carolina State University, <http://www.ie.ncsu.edu/kay/matlog/>
 90. MESAROVIC, M.D., MACKO, D., TAKAHARA, Y., “*Theory of Hierarchical, Multi-Level*

- Systems*,” Academic Press, New York and London, 1970
91. MILLER, B., CLARKE, J-P., “*Emissions Reductions Enabled by Improved Airport Operations*,” Air Traffic Control Quarterly Journal, Volume 12, Number 2, 2004
 92. MIT Encyclopedia of Cognitive Sciences:
<http://cognet.mit.edu/MITECS/Entry/richardson.html>
 93. MORRELL, P., CHERIE, H.-Y. LU, “*Social Costs of Aircraft Noise and Engine Emissions*,” Transportation Research Record, n 1703, p. 31-38, 2000
 94. MORIN, M., “*Metrics and Methods for Improving Airline Schedule Reliability*,” Master’s Thesis, Department of Aeronautics & Astronautics, Massachusetts Institute of Technology, September 2001
 95. PRASAD, B., “*Concurrent Engineering Fundamentals*,” Volume II, Prentice Hall, NJ, 1996
 96. PSARAFTIS, H. N., “*A Dynamic Programming Approach to the Aircraft Sequencing Problem*,” Ph.D. Thesis, Flight Transportation Laboratory Report R78-4, M. I. T., Cambridge, MA, 1978
 97. PUJET, N., “*Modeling and Control of the Departure Process of Congested Airports*,” Ph.D. Thesis, Department of Aeronautics and Astronautics, M. I. T., Cambridge, MA, September 1999
 98. PUJET, N., FERON, E., “*Input-Output Modeling- and Control of the Departure Process of Congested Airports*,” Air Traffic Control Quarterly Journal, Volume 8, Number 1, pp. 1 – 32, 2000
 99. RICHARDSON, J., “*Departure Sequencing Engineering Development Model Joint Applications Development Methodology*,” 1995 Annual Air Traffic Control Association Convention Proceedings, 1995, pp. 143 – 147
 100. RICHETTA, O., “*Ground-Holding Strategies for Air Traffic Control Under Uncertainties*,” Ph.D. Thesis, Massachusetts Institute of Technology, June 1991
 101. RICHETTA, O., ODoni, A.R., “*Solving Optimally the Static Ground-Holding Policy Problem in Air Traffic Control*,” Transportation Science, Vol. 27, No. 3, pp. 228-238, August 1993

102. RINDERLE, J.R., "*Measures of Functional Coupling in Design*", Ph.D. Thesis, Massachusetts Institute of Technology, June 1982
103. ROBINSON, J.E. III, ISAACSON D.R., "*Development of A Closed-Loop Testing Method for a Next-Generation Terminal Area Automation System*," Proceedings of the American Control Conference (ACC), Anchorage, AK, May 8-10, 2002
104. ROGERS, J.L., BLOEBAUM, C.L., "*Ordering Design Tasks Based on Coupling Strengths*," Proceedings, 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, 1994, AIAA 94-4326-CP, pp. 708-717
105. SADOUNE, M.M., "*Terminal Area Flight Path Generation Using Parallel Constraint Propagation*," Ph.D. Thesis, Flight Transportation Laboratory Report R89-1, M. I. T., Cambridge, MA, 1989
106. SAKSENA, V. R., O' REILLY, J., KOKOTOVIC, P. V., "*Singular Perturbations and Time-Scale Methods in Control Theory: Survey 1976-1983*," Automatica, Vol. 20, No. 3, May 1984
107. SCHUBERT, M., "*Arrival and Departure Management within the Terminal Area*," AIAA 96-3720, Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA, San Diego, CA, July 29-31 1996
108. SCHULER, M., P., "*Reducing the Complexity of the Software Design Process with Object-Oriented Design*," NASA Langley Research Center, Hampton, VA, available at <http://techreports.larc.nasa.gov/ltrs/PDF/conf-tech2001-v1.pdf>
109. SHUMSKY, R.A., "*Dynamic Statistical Models for the Prediction of Aircraft Takeoff Times*," Ph.D. Thesis Operations Research Center, MIT, Cambridge, MA, 1995
110. SHUMSKY, R.A., "*Real-Time Forecasts of Aircraft Departure Queues*," Air Traffic Control Quarterly Journal, Volume 5, Number 4, 1997, pp. 281-308
111. SOLBERG, I., "fminconset: solves constrained minimization problems, when some of the variables are restricted to discrete values", Matlab File Exchange, <http://www.mathworks.com/matlabcentral/fileexchange/index.jsp>
112. SPENCER, A., SMITH, P.J., BILLINGS, C., "*Surface Management System Simulation*:"

- Findings Regarding the TMC Position*,” Institute for Ergonomics Technical Report #2002-2, Ohio State University, Columbus, OH
113. SPENCER, A., SMITH, P.J., BILLINGS, C., BRINTON, C., ATKINS, S., “*Support of Traffic Management Coordinators in Airport Air Traffic Control Towers Using a Surface Management System*,” In S. Chatty, J. Hansman and G. Boy, Proceedings of the International Conference on Human-Computer Interaction in Aeronautics, Cambridge, MA: AAAI Press, 105-111
114. STEWARD, D.V., “*The Design Structure System: A Method for Managing the Design of Complex Systems*,” IEEE Transactions on Engineering Management, Vol. EM-28, No.3, pp. 71-74, 1981
115. STRANG, G., “*Introduction to Applied Mathematics*,” Wellesley – Cambridge Press, 1986
116. SUH, NAM P., “*Axiomatic Design: Advances and Applications*”, New York, Oxford University Press, 2001
117. Surface Management Systems (SMS) research summaries:
- a) http://ffc.arc.nasa.gov/our_projects/sms.html
 - b) <http://www.metronaviation.com/airport-surface-mgt.html>
118. TEIXEIRA, R., “*An Heuristic for the Improvement of Aircraft Departure Scheduling at Airports*,” Ph.D. Thesis, Loughborough Technical University, March 1992
119. TERRAB, M., ODoni, A.R., “*Ground-Holding Strategies for ATC Flow Control*”, Ph.D. Thesis, Department of Civil Engineering, Massachusetts Institute of Technology, February 1990
120. VAN LEEUWEN, P., HESSELINK, H., ROHLING, J., “*Scheduling Aircraft Using Constraint Satisfaction*,” 11th Workshop on Functional and Constraint Logic Programming, Grado, Italy, June 20-22, 2002
121. VENKATAKRISHNAN, C. S., “*Analysis and Optimization of Terminal Area Air Traffic Control Operations*,” Ph.D. Thesis, Operations Research Center, M. I. T., Cambridge, MA, February 1991
122. VENKATAKRISHNAN, C. S., BARNETT, A., ODoni, A.R., “*Landings at Logan Airport*:

- Describing and Increasing Airport Capacity*,” *Transportation Science*, Vol. 27, No.3, pp. 211-227, August 1993
123. VELEZ-REYES, M., “*Decomposed Algorithms for Parameter Estimation*,” PhD. Thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, September 1992
124. VIOLETTE, J. D., “*Implementation of a Hierarchical Controller in a Factory Simulation*,” M.Sc. Thesis, Department of Aeronautics & Astronautics, M.I.T., Cambridge, MA, June 1993
125. VÖLCKERS, U., BROKOF, U., DIPPE, D., SCHUBERT, M., “*Contributions of DLR to Air Traffic Capacity Enhancement within a Terminal Area*,” AGARD meeting on ‘Machine Intelligence in Air Traffic Management’, May 1993
126. VÖLCKERS, U., BÖHME, D., “*Dynamic Control of Ground Movements: State-of-the-Art Review and Perspectives*,” AGARD-R-825, 1997
127. VRANAS, P., “*The Multi-Airport Ground Holding Problem in Air Traffic Control*,” Ph.D. Thesis, Operations Research Center, M. I. T., Cambridge, MA, June 1992
128. VRANAS, P., BERTSIMAS D., ODoni, A.R., “*The Multi-Airport Ground-Holding Problem in Air Traffic Control*”, *Operations Research*, Vol. 42, No. 2, pp. 249-261, 1994
129. VRANAS, P., BERTSIMAS D., ODoni, A.R., “*Dynamic Ground-Holding Policies for a Network of Airports*”, *Transportation Science*, Vol. 28, No. 4, pp. 275 – 291, November 1994
130. WAGNER, T.C., PAPALAMBROS, P.Y., “*A General Framework for Decomposition Analysis in Optimal Design*,” *Advances in Design Automation – 1993*, B.J. Gilmore (ed.), Vol.2, ASME, New York, pp.315-325, 1993
131. WAGNER, T.C., PAPALAMBROS, P.Y., “*Implementation of Decomposition Analysis in Optimal Design*,” *Advances in Design Automation – 1993*, B.J. Gilmore (ed.), Vol.2, ASME, New York, pp.327-335, 1993
132. WILKINSON, T., “*Departure Manager Stakeholder Requirements Survey, Airport Visits: Summary*,” Report (Released Issue) of the Eurocontrol European Organization for the

References

- Safety of Air Navigation, August 1999, available at <http://www.eurocontrol.int/odt/documents/library.html>
133. WISMER, D.A., “*Optimization Methods for Large-Scale Systems*,” McGraw-Hill Company, 1971
134. WONG, G.L., “*The Dynamic Planner: the sequencer, scheduler and runway allocator for air traffic control automation*,” report NASA/TM-20000-209586, NASA Ames Research Center, Moffet Field, CA, April 2000, available at http://www.ctas.arc.nasa.gov/publications/papers/wong_04_00.pdf

Appendix A

FIELD OBSERVATIONS

The development of decision support tools for air traffic controllers calls for a thorough understanding of links and interactions in ATM operations and requires constant evaluation and assessment. In addition, the design of a high-level architecture for airport departure management should be based on a thorough analysis of the airport system and understanding of the needs and constraints in current airport operational procedures. To this end, a significant set of field observations was done at Boston Logan International Airport [71], [72], [73].

A.1 The Departure Process –Field Observations

Analysis of departure and arrival operations at Logan (BOS) and other major US airports, such as Chicago O'Hare (ORD), Atlanta Hartsfield (ATL) and Dallas-Fort Worth (DFW) revealed significant operational delays and environmental impacts associated with the departure process

[71]. Analysis also revealed that, while there are many similarities between the departure and arrival processes, there are also significant differences, which affect the way in which improvements may be effected. For example, beyond a certain entry fix point in the terminal airspace, the arrival stream is fixed and there is not much opportunity for sequence adjustments. On the ground, however, while there is little observability and high volatility associated with departure operations [72], [73], controllers have more opportunity to affect the final runway operations sequence and timing. Therefore, given that, for safety reasons, controllers typically prefer to keep aircraft on the ground rather than in the air, designing a decision-aiding system to assist controllers in handling and optimizing departure operations on the airport surface can be beneficial to ATM operations.

Different components of the airport were identified as flow constraints, which introduce delays and inefficiencies and contribute to the low prediction capability associated with departures [71], [72], [73]. Some examples of the inefficiencies identified through the field observations at Logan are given in section A.2. The flow constraints identified were associated with the main airport system elements:

- a) The gates complex
- b) The ramp area
- c) The taxiway system and
- d) The runway system

The flow constraints manifest through the aircraft queues that physically form at each element.

In that sense, an airport system can be modeled as a complex interactive queuing system in which departures and arrivals are highly coupled. All the different types of aircraft queues that form on the Logan airport surface in the configuration 22L / 22R / 27 are illustrated in Figure A.1 and Figure A.2.

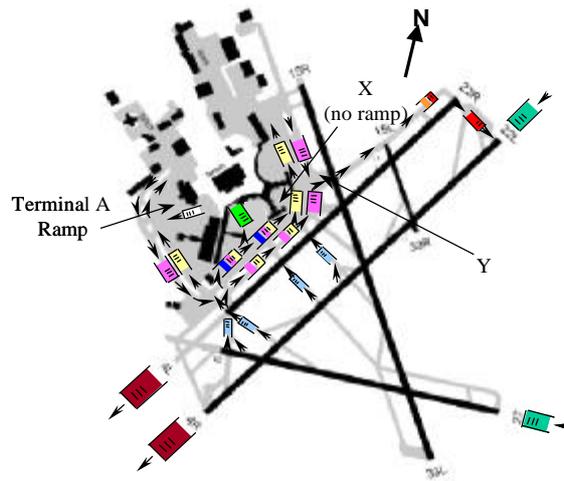


Figure A.1: Taxiway and Runway Queues at Logan Airport under configuration 22L / 22R / 27

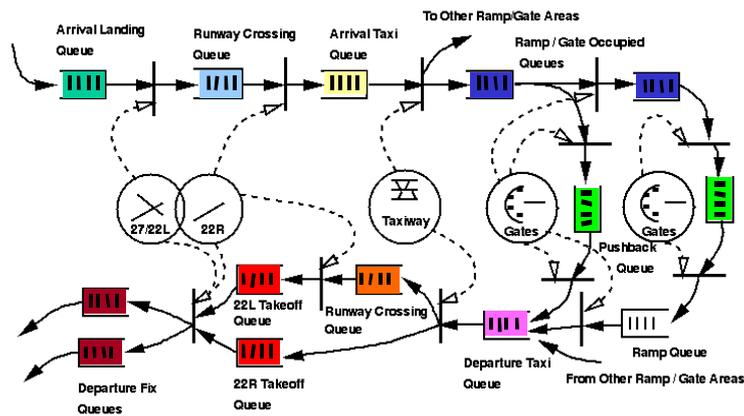


Figure A.2: Queuing Model for Logan Airport under configuration 22L / 22R / 27 (source [74])

As shown in Figure A.1, in this configuration, runway 27 is usually dedicated to arrivals, runway 22R is used only for departures and runway 22L is used primarily for arrivals. Often, pilots who specifically request a longer runway for takeoff, use the latter for departure, in which case they line up and wait on the north taxiway segment between 22L and 22R. When a large number of departures are expected, the airport switches to "Accelerated Departure Procedures" (ADP), in which case runway 22L is used only for departures and all arrivals are routed to runway 27.

The flow of arriving and departing aircraft through the airport system and the various queues forming on the airport surface in configuration 22L / 22R / 27 can be abstracted as in Figure A.2. The physical elements of the airport system (gates, taxiways, runways) are depicted in the middle part of the figure and their interactions with the airport queues are shown as dashed arrow lines. Each bar between different queues represents a transition from one queue to another. Solid arrows represent the aircraft flow and dashed arrows associated with a specific airport resource represent use of that resource for the queue transitions. Following the aircraft flow in Figure A.2, an arriving aircraft queues on final approach and after landing on runways 27 or 22L, it joins a runway-crossing queue waiting to cross runway 22R. After crossing, it joins other aircraft in taxiing queues, which include arriving and/or departing aircraft. Upon arrival at its assigned gate, it may have to wait for the gate to be released from the previous aircraft. When the gate becomes available, the aircraft joins a pushback queue according to its scheduled departure time. The two different types of gates that were observed are depicted in Figure A.2. In one case (far right side of Figure A.2) aircraft that push back from these gates enter the ramp area (e.g. Logan terminal A ramp in Figure A.1⁴⁶) and wait in a ramp queue for ATC clearance

⁴⁶ Referring to the time when terminal A still existed at Logan airport and after completion of the new terminal A is completed.

to enter the departure taxi queue in the taxiway system. In the other case, aircraft push back directly onto the taxiway system with no intermediate ramp area (e.g. point X in Figure A.1). At point Y, departing aircraft that are assigned to runway 22R for takeoff enter the 22R takeoff queue. Departing aircraft assigned to take off on runway 22L, join the same taxiway segment but are considered to enter a runway-crossing queue in order to cross runway 22R before joining the 22L takeoff queue.

The possible location of these queues on the surface of Logan Airport in the case of configuration 22L / 22R / 27 is shown in Figure A.1. The two arrival queues on runways 27 and 22L are easily identified, as well as the departure queue that is formed on the taxiway segment adjacent to runway 22R. This departure queue includes aircraft that line up to take off on runway 22R and aircraft that will cross 22R to take off on 22L (ADP). Operations on runway 22R are impeded not only by aircraft departing on 22L but also by arriving aircraft that queue in taxiway segments between the two parallel runways in order to cross runway 22R.

The above identification of the airport flow constraints and the associated queuing network was critical in studying the dynamics of departure operations. It enabled the definition of various control points where the departure operations could be affected by control actions and it also helped in determining the control options of the Departure Planner decision-aiding tool.

The observed operations described above are tactically under the control of the air traffic controllers in the Control Tower. Air traffic controllers have to clear each aircraft to use an

airport resource such as a ramp, a taxiway or a runway, by delivering a “clearance instruction”. By observing the aircraft queuing process and the control process, it was possible to identify control functions, which occur at specific control points. The transitions in Figure A.2 are examples of such control actions and control points. Most aircraft operations are under the control of the Tower and therefore, control actions and points are associated with the controllers’ clearances. However, some aircraft operations, such as engine-start, are not currently under the Tower’s control at Logan.

The following control functions were identified based on the observations at Logan airport:

- a) Pushback clearance (for jets) or taxi clearance (for propeller aircraft).
- b) Clearance to enter the taxiway system from the ramp area, gate where the aircraft is waiting.
- c) Runway and taxi-path allocation, i.e. the process of routing aircraft to a specific runway, through a predetermined taxiway path.
- d) Sequencing of aircraft takeoff, i.e. merging of aircraft into the same takeoff queue or mixing between aircraft from multiple queues (e.g. one jet aircraft and one turboprop aircraft queue) at the same takeoff runway and
- e) Takeoff release of each aircraft. If a runway is used by departures, landings and runway crossings, takeoff release also involves mixing of operations on that runway.

It was observed that a control function could be exercised at different times and locations. For example, aircraft sequencing can be performed at the gate (pushback control), at the taxiway entry points as aircraft are released into the taxiway system and up to the physical point beyond

which the aircraft have to commit to a particular takeoff queue. Once the aircraft are physically present at the runway end, the takeoff sequence is hard to modify. Therefore, notionally, a control point is defined as *“the last opportunity that the controllers have to apply a particular control function to the departure queues”*. A control point can be a physical point on the airport surface, or it can be a point in time during the departure process, when the aircraft transitions from one state to another. For example, a control point exists at the gates when aircraft are cleared to push back into the ramp area. A possible control point also exists at the instant when aircraft, while taxiing, are handed off to a specific controller who handles a particular set of runways. At that point in time these aircraft are committed to enter a runway queue, with much less room for further adjustments than in the taxiing phase. The main control points associated with the observed control functions outlined above are:

- a) The gate.
- b) The point of entry from the gate or ramp into the airport taxiway system.
- c) The point of commitment to a specific queue (temporal or spatial)
- d) The point of entry to an active runway and
- e) The point of takeoff release (exit from a takeoff queue).

The control points and the control functions are generic but there are differences that are airport-specific and runway configuration-specific. In [71] Atlanta Hartsfield (ATL) and Boston Logan (BOS) were compared. BOS has one runway system and minimal ramp area, while ATL has two runway systems and a well-defined ramp area. Therefore, BOS is usually operating with one departure queue. The structure of this queue can be primarily determined at the gate (pushback

clearance control function), which in most cases is the point of entry into the taxiway system, since the intermediate ramp area in BOS is almost non-existent. On the other hand, ATL has at least two departure queues in most cases, as well as a larger controlled ramp area than BOS does. This means that the structure of the departure queues can be affected at the gate control point, but also to a larger extent at the taxiway entry points and through the mixing of aircraft from different queues.

A.2 Operational Inefficiencies Observed

The following analytical examples of airport surface operational inefficiencies are extracted from observations of actual operations at Logan airport. They are used to describe actual inefficiencies in runway utilization due to Air Traffic Management operational constraints and also demonstrate the potential benefits that the airport system can have from efficient utilization and planning of the runway resources. Actual operations at Logan airport under configuration 22L / 22R / 27 are presented in Figure A.3 and Figure A.4 where the progress of five departing flights through time is followed. From left to right, the following time events are given:

- a) Scheduled departure time
- b) Time the pilot called “ready” to leave the gate
- c) Time the clearance to pushback (for jets) or taxi (for props) was granted by the tower and
- d) The “Monitor Tower” time which (in this configuration) is the time that each departing aircraft is cleared to join the takeoff queue. This is the point where controllers usually exercise sequencing control among departures.

The two remaining timelines in each of the two figures describe the takeoff and crossing operations on runway 22R (presented once in scale and once magnified). On these timelines, each departure and crossing event has a record of the time the aircraft was cleared to take off or cross runway 22R.

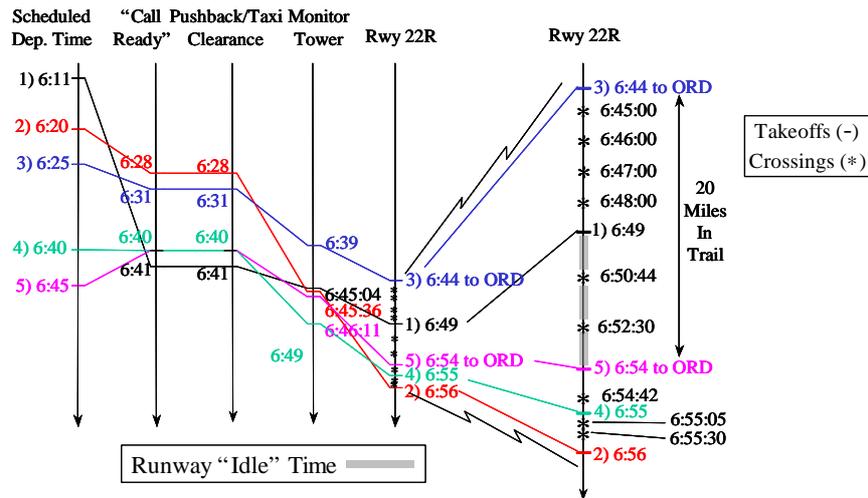


Figure A.3: Miles In Trail (MIT) restrictions

In Figure A.3 Chicago-bound flights (ORD) were under a 20 Miles-In-Trail restriction. Hypothetically, departure 1, which called ready to push back at 6:41 am after departure 5 (6:40 am) was sequenced to take off before the latter, so that ORD-bound departure 5 maintains a 20 MIT separation from prior ORD-bound departure 3. Departures 2 and 4, which called ready for pushback before or together with departure 5, ended up behind the latter in the takeoff sequence. Flights 2 and 4 could have been sequenced to take off before flight 5, which was already delayed due to the MIT restriction, resulting in more efficient runway throughput. Note that in a 5-minute period between flights 1 and 5, the runway was used only for two aircraft crossings

(asterisk “*” symbols on the runway 22R timeline). The data collected showed that there were:

- A “runway idle” time gap of 104 sec, between departure 1 and the first crossing event that occurred immediately after it (6:49 to 6:50:44)
- A gap of 106 sec (6:50:44 to 6:52:30) between the two crossings and
- A gap of 90 sec (6:52:30 to 6:54) between the second crossing and the next departure 5.

Looking at the utilization of runway 22R, it is suggested that there is room for assisting air traffic controllers in designing takeoff sequences under several operational constraints, with an objective to avoid such unnecessarily large runway utilization “gaps” and achieve higher takeoff throughput.

Similarly, in Figure A.4, we can observe large time gaps between successive takeoffs with only one crossing operated within each gap. For simplicity, the “call ready” timeline has been omitted in this case. The five departures examined here call ready and pushback in the order 1, 3 & 4, 2, and 5. Departure 5 called ready for pushback (9:18) after departure 2 (9:09). However, it was sequenced to take off before the latter, probably due to the fact that, in this configuration, the estimated taxi out time for flight 5 is lower than the one for flight 2, based on the distance from each flight’s originating terminal to the runway 22R takeoff point. From this example, it is evident that controllers actually implement certain sequencing rules, based primarily on their operational experience. Nevertheless, there seems to be room for improved sequences, in order to avoid excessive “runway idle” time. Focusing on the takeoff / crossing timeline in Figure A.4, we can see that in each of the four time intervals between the five departures, there was only one crossing operated. Each crossing occurred very soon after the preceding takeoff and for the

remaining time until the next takeoff the runway remained idle. Based, on the wake vortex separation requirements for the aircraft types present, only the gap between departures 1 and 4 needed to be as long as it actually was.

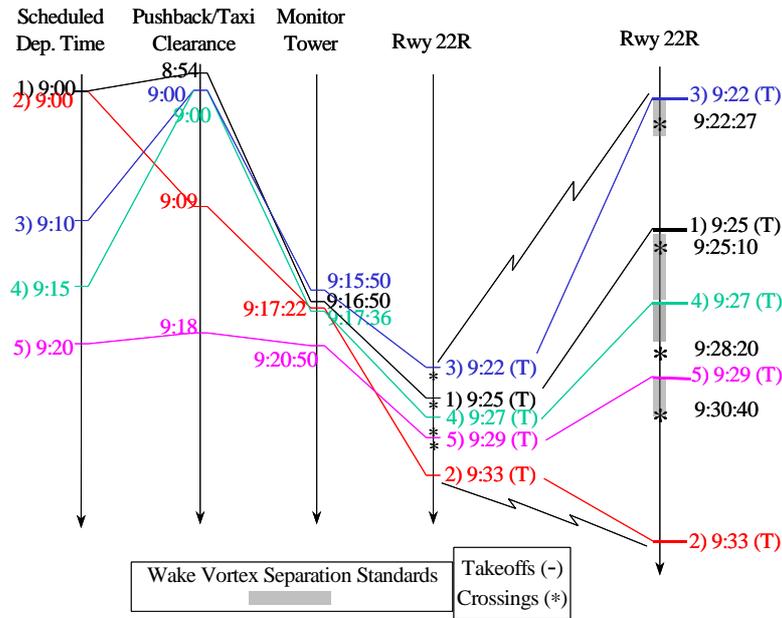


Figure A.4: Takeoff sequence adjustments

Note that, even if a takeoff sequence is well designed based on the latest information available to the controllers, there are always unexpected events that can cause serious deviations from the planned takeoff order. For example, departure 2 may have been carefully sequenced after departure 5, as mentioned above. However, it was cleared to take off 140 sec (9:30:40 to 9:33:00) after the previous crossing aircraft cleared the runway and 4 min (9:29 to 9:33) after the previous takeoff occurred on the same runway. This possibly happened because the takeoff weight and balance calculations were not available for departure 2, even though it was first in the takeoff queue. In other cases, scheduled or non-scheduled priority flights (e.g. lifeguard flights)

or aircraft in the departure queue unable to takeoff can also disrupt the takeoff sequence.

Appendix B

CONCEPTUAL ARCHITECTURE OF A DEPARTURE PLANNING DECISION-AIDING SYSTEM

Field observations, as a method of airport system identification, helped in identifying the control points and functions mentioned in Appendix A. The results from this analysis were combined with documented airport and Air Traffic Control (ATC) operations in order to generate the system architecture presented here. This conceptual architecture provides an initial idea of how an airport surface operations planning and control system could be introduced in a real-world airport operational environment. The two principal parts that the architecture consists of are illustrated in Figure B.1:

Strategic Planner

This is essentially a Configuration Planner that would typically have an approximately 3 to 4-hour time horizon. It would perform configuration management tasks by attempting to match the airport’s operational capacity as closely as possible to the scheduled demand, always taking under consideration limitations imposed by the forecasted terminal weather and by environmental constraints, such as noise restrictions. More details on the functionality of the Configuration Planner are given in section B.1.

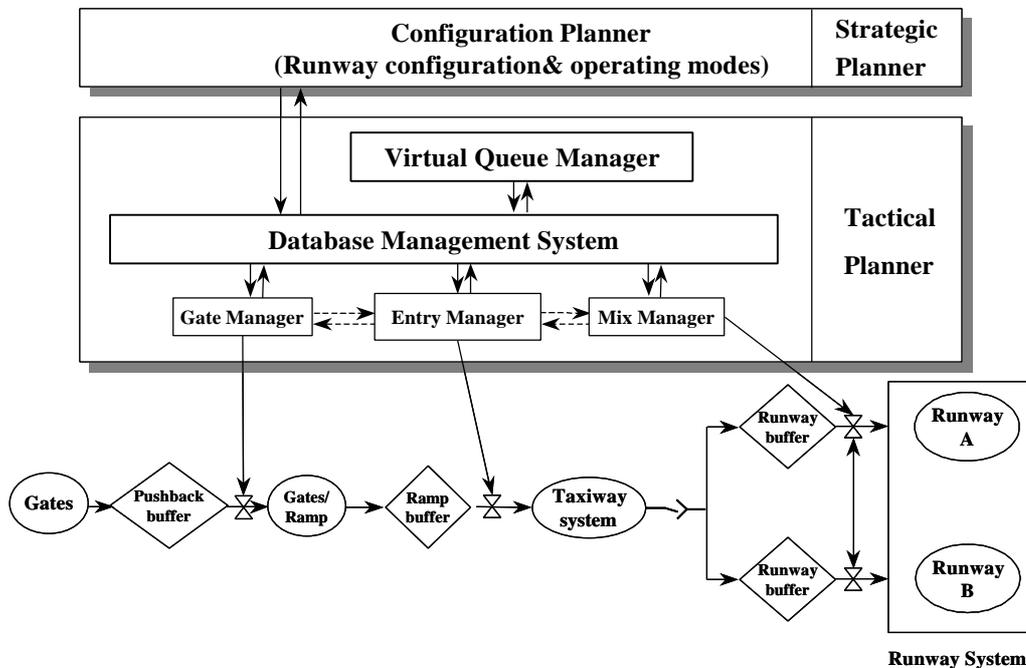


Figure B.1: Conceptual Architecture for the Departure Planner

Tactical Planner

This part of the system has an approximately 15-30 minute time horizon, performs tactical planning of runway operations under a specific runway configuration and exercises appropriate

control to implement the generated plans.

The most critical tactical component introduced in the system is the *Virtual Queue Manager*.

The remaining three tactical system components (starting from the gates and following the departure flow to the runway takeoff queues) are:

- a) The Gate Manager, which is introduced in order to support the controllers in managing the pushback schedule given the unpredictability (uncertainty) inherent in airline gate operations and schedules.
- b) The Taxiway Entry Manager, which modulates the release of aircraft for entry into the taxiway system and
- c) The Mix Manager, which is introduced in order to manage the arrival/departure mix onto active runways.

Detailed descriptions of the various Departure Planner components are provided in the following sections. In a generic framework that can be applied to any airport, each of the three tactical components is designed to exercise control and address inefficiencies at specific control points along the departure process. Each strategic and tactical component can be linked (mapped) to a group of objectives drawn from the family of the general functional objectives of the decision-aiding system that are presented in Chapter 3, as well as in Figure C.1 of Appendix C. At the same time, all components are envisioned to communicate and exchange data with each other directly or via a common Database Management System (Figure B.1). The latter is designed to ensure that all components have access to the same consistent information. It should interact with several specialized databases containing necessary airport specific data, such as:

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

- Airport topology
- Aircraft performance data⁴⁷, as well as dynamically generated data, such as:
 - Flight plans and
 - Aircraft identification information
- ATC procedures, regulations and restrictions, e.g. airport-specific arrival and departure routes, such as Standard Terminal Arrival Routes (STAR) and Standard Instrument Departures (SID) and
- ATC constraints that are dynamically introduced, e.g. flight priorities, Ground Delay Programs

Generalizing, the main tasks of all components should be to:

- Implement the designed surface operations plan through control actions, taking into account:
 - Specific information and constraints for the area of operations that each component is associated with, i.e. gates, ramp, taxiways, runways and
 - Specific “settings” input in the system by the controller responsible for that area
- Distribute information
 - To and between airport operators, i.e. the controllers and airport users such as the airlines (external interface). An example of such external communication is the distribution of information to controllers that cannot be directly accessible by them, but is contained in

⁴⁷ From available databases, such as Eurocontrol’s Base of Aircraft Data (BADA).

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

other, in some cases remote, technical systems, such as central airline operations' management systems, or local airline station control networks and

- In between system components (internal interface). Examples of such internal communication is the distribution of information about all performed control actions of a specific component, which are relevant to other components of the architecture, or the communication of new events that make certain system solutions infeasible.

In any busy airport, even with a fairly simple runway geometry, managing ground operations and planning the allocation of runway time to the various types of runway operations, can be a very challenging task. Any effort to design a decision-aiding tool deployed to assist air traffic controllers with this task must take into consideration several operational requirements, which arise primarily due to the presence of human operators in the process. For example, in managing airport ground operations, there are two main tasks to be performed: planning and control and both of them are inevitably performed in a distributed fashion, because of the many different parties involved in most planning and control functions. Therefore, a successful design of an automated decision-aiding system that will be under airport-wide use must take into account integration issues between the varying objectives, constraints, information requirements and control inputs from all stakeholders involved in airport ground operations, such as the airlines, airport authorities, airport service operators, passengers and the FAA personnel in the air traffic control tower and the TRACON room.

As an example, consider the information needs of the gate personnel and station managers of a

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

specific airline at an airport, as opposed to the information needs of the air traffic controllers in the tower cab. The latter have to deal with all incoming and outgoing airport ground traffic and not only the aircraft of a particular airline, which means that their information needs are apparently larger than those of the specific airline operators. In addition, the controllers' objectives are naturally more system-oriented than the understandably profit-oriented objectives of an airline operations control center.

It is uncertain how feasible it is for an automated system to take into account the objectives of all involved parties. Therefore, it is likely that the solutions suggested by such a system may not always be optimal. In addition, runway operations plans generated by the decision-aiding tool may even be unimplementable due to its inability to incorporate very dynamic last minute changes of the system state (e.g. traffic situation) that it was not aware of, when those plans were calculated. Real-world applicability of the generated plans requires that the controllers have ample planning and control flexibility to modify the plans for factors and events not visible to the decision-aiding tool and that the solutions generated are presented to controllers as operational guidelines and not as inviolable laws that they have to abide to.

On a final note, careful consideration must be given to the interaction between the controller and the decision-aiding tool in the context of the operational environment and standard procedures within the ATC tower. The underlying structure of the decision-aiding and optimization processes must be clear to the controller and the results must be consistent with the controllers' heuristic estimates in order to ensure the system's "usability" by the operators. In addition, the interface must limit "head-down" time for controllers. It should also be noted that such a

decision-aiding tool could be used as a communication device between controllers. Controller interface devices, which support these functions in the tower cab environment, are under development.

B.1 Configuration Planner

Functionality

The main task performed by the Configuration Planner is the development of the runway configuration plan for the airport so that all arrivals and departures expected to utilize the airport runway resources can be handled. It must be designed to take into account the stochasticity associated with weather. Accurate terminal weather and wind forecasts in conjunction with the pertinent noise abatement rules are used to define the set of feasible configurations for the airport. Based on these, the Configuration Planner determines the number of hourly operations that the airport can handle and the expected arrival and departure demand over successive intervals during the planning horizon is then matched to each of these configurations (level 1 in Figure B.2), in order to design the best configuration strategy throughout the day.

In order to accommodate short-term demand fluctuations (Figure B.4) the Configuration Planner should be able to suggest discrete operating modes within the time horizon of each of the planned configurations (level 2 in Figure B.2), especially since the response time of the airport to a change from one operating mode to another within the same runway configuration is quicker than the airport response time in the case of a full change of runway configuration.

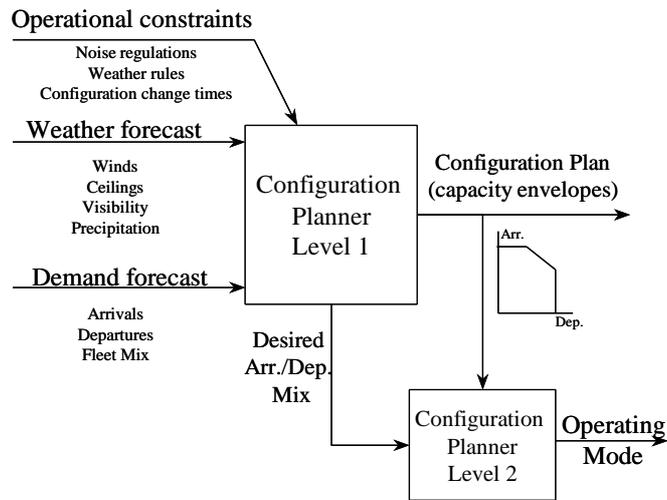


Figure B.2: The Configuration Planner

Supporting Analysis

Examples of four capacity envelopes that were generated based on airport throughput data over the course of 15 days are presented in Figure B.3. The data was collected from the air traffic control tower records as well as ETMS records from the FAA’s CODAS database and the analysis was first presented in [74]. Evidently, different runway configurations at Logan airport yield different capacities. In their effort to match the levels of demand expected, the controllers try to use the configurations with the highest arrival and departure capacity. However, the final selection is dependent on the weather and wind conditions at the airport. In addition, there are several noise abatement rules that limit the use of certain runways at certain times of the day. These weather and noise constraints are taken into account by the controllers in their decision-making process for selecting the airport runway configuration. In fact, at Logan airport, there are certain runway configurations that are recommended at night hours.

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

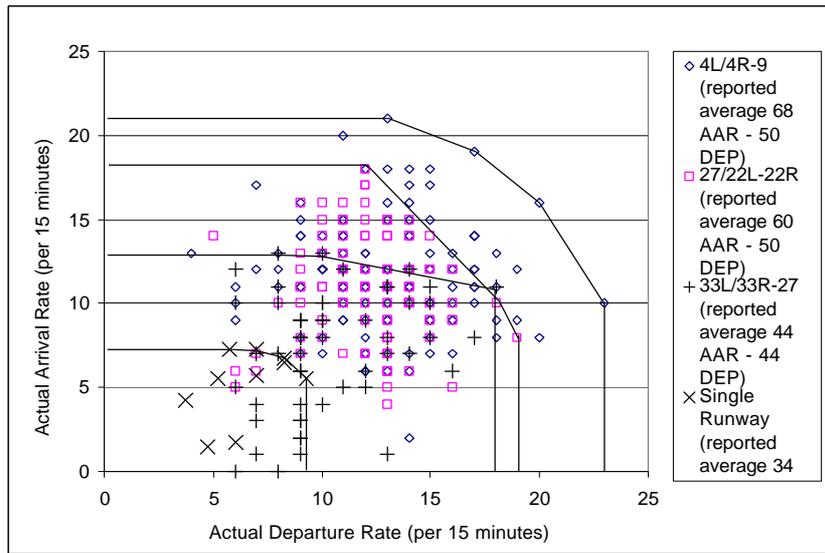


Figure B.3: Runway Configuration Capacity Envelopes (Graph source: [74], Data source: ETMS / Tower Records, 7-9 AM, 4-8 PM, July 1-15 1998 except Saturdays, Logan Airport)

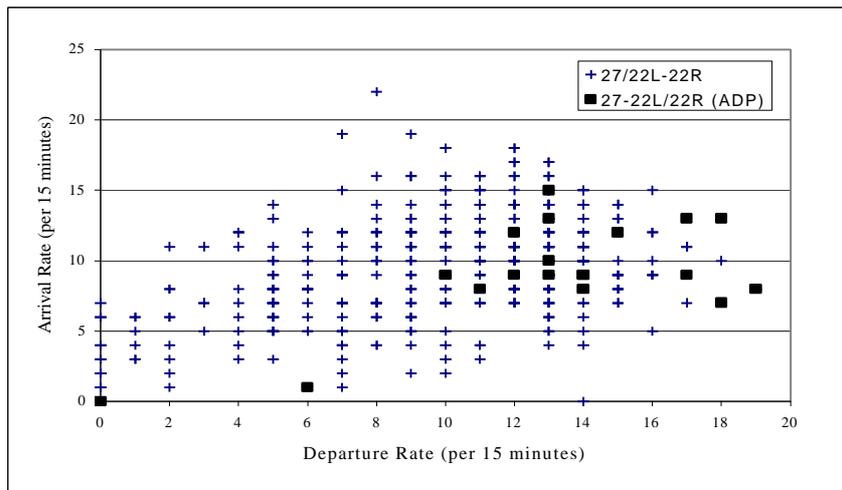


Figure B.4: Accelerated Departure Procedure (ADP) (Graph source: [74], Data source: ETMS, July 1-15 1998, Logan Airport)

Matching different possible configurations to the schedule takes into account the time required for transitioning between configurations, which can take values up to 20 min for a busy period at major airports like Logan. When the arrival flow is very high, it takes longer to implement a

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

configuration change, because it is harder to interrupt the arrival stream on final approach. For example, at Logan airport, switching to a high capacity configuration is usually attempted before periods of expected high traffic.

Short-term fluctuations in the arrival/departure mix drive the airport in “departure push” or “arrival pull” mode. In these cases, the air traffic controllers perform short-term configuration changes by adjusting the operations that are assigned to utilize each runway within the current configuration. These configuration changes correspond to transitions between different operating points on the airport’s capacity curve [73]. For example, in normal operations within the 22L / 22R / 27 configuration, runway 22L is used both for arrivals and departures. However, when Logan airport is in a departure push mode, runway 22L is sometimes used only for departures (together with 22R) and all arrivals are assigned to runway 27. The effect on the departure / arrival mix is shown in Figure B.4, by superimposing periods when ADP was used, over the capacity envelopes presented in Figure B.3 [74]

In matching the scheduled demand to the set of possible configurations, the configuration planner should take into account the uncertainty inherent in departure operations. Departure demand is affected by airline decisions on delays and cancellations, which are not always known sufficiently in advance. Collaborative Decision-Making (CDM) is a step towards addressing this problem. It has been shown that advance cancellation notices have improved noticeably after the introduction of CDM [14].

B.2 The Virtual Queue Manager

The Virtual Queue Manager (VQM) is envisioned to proactively manage the airport's Virtual Queue so that the Departure Planner objectives are met and therefore airport resources (runways, taxiways and gates) are efficiently utilized.

Virtual Queue Definition

A *Virtual Queue* can be defined as a notional waiting line of departing aircraft arranged, at any instant of time, according to the order in which they are expected to take off. It consists of:

- a) A “physical” part, which involves aircraft that are or will shortly be physically present at a certain location on the airport surface, but for which there is no further chance for re-sequencing; therefore, these aircraft have a fixed (“frozen”) position in the virtual queue and
- b) A “virtual” part, which involves aircraft that are scheduled to occupy a particular position in the takeoff sequence of aircraft, but are neither physically present in the takeoff queue yet nor are they “frozen”. Position assignments in this virtual part of the virtual queue are very much subject to revision.

In other words, the Virtual Queue can be seen as an extension of the notion of a physical queue that depicts the final takeoff sequence of all scheduled departures as the Departure Planner has planned it up to the current point in time.

If two or more departure runways are in use, then multiple virtual queues (one for each departure runway) can be used. As an alternative, in such cases there might be a single virtual queue with

each aircraft in the queue being “tagged” to indicate which departure runway it will use.

Functionality

In Figure B.1, the VQM is hypothesized to reside in the system hierarchy at one level above the tactical Departure Planner elements. It interacts separately with the strategic configuration planner and also acting as a central processing function it coordinates the three tactical Departure Planner components. It incorporates all the requests from various physical queues in the system and relays back to them information about generated runway operation plans, the virtual queue and the required control actions in order to implement it. The major challenge is to design the optimum size of the virtual queue (minimum buffer size) in such a way that the aircraft queues in the system (especially the runway takeoff queue) are not consistently “starved” or saturated.

Since the runway was observed to be the main flow constraint, a possible design of the virtual queue is generated assuming that its “physical” part resides at the runway threshold. In this case, the two parts of the virtual queue would be:

- a) The “physical” part, including flights whose position in the queue may be “frozen” a few (10 or 15) minutes before their assigned takeoff time and
- b) The “virtual” part, in which the scheduled departure time and the sequencing of some aircraft may be subject to change due to the fact that there is still considerable time to go, e.g., more than 15 minutes until the actual departure event.

The mode of interaction between the Virtual Queue Manager (VQM) and the other Departure

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

Planner tools is still a research issue. One possibility is a "Master-Slave" relationship, in which the optimization logic is entirely included in the Virtual Queue Manager. Each of the Departure Planner components simply relays information and communicates its specific requests to the VQM with the hope that the system status will allow its requests to be satisfied. Another possible design philosophy is for each of the Departure Planner components to carry its own optimization logic and perform a local optimization dealing with a specific subproblem of the overall problem. Subsequently, the Virtual Queue Manager takes all the individual optimization results from the Departure Planner components and attempts to combine all the "local" solutions into a "global" one. This process may involve iterations and re-optimization until a feasible solution is achieved.

A hypothetical means to visualize the virtual queue and understand its potential benefits is provided in Figure B.5. Each side of the figure represents a snapshot of the takeoff sequence as it is currently projected in the future. The "left hand" side represents the observed behavior where aircraft are transitioned from one state to the next mainly in a First Come First Serve order and where the queue buffer sizes are not controlled resulting in unnecessarily overloaded takeoff queues ("Frozen") and taxiway congestion.

The behavior with the implementation of the Virtual Queue, which provides a tool for effectively controlling the number of aircraft in each state at each point in time and regulating the timing of aircraft transitions from one state to the next is represented in the "right hand" side in Figure B.5. Aircraft move from the gate to the ramp onto the taxiway system and into one of the takeoff queues, following the sequence and timing commanded by the Virtual Queue Manager. This

optimal (or near-optimal) sequence is determined based on the system-wide objectives and constraints that were discussed in Chapter 3⁴⁸.

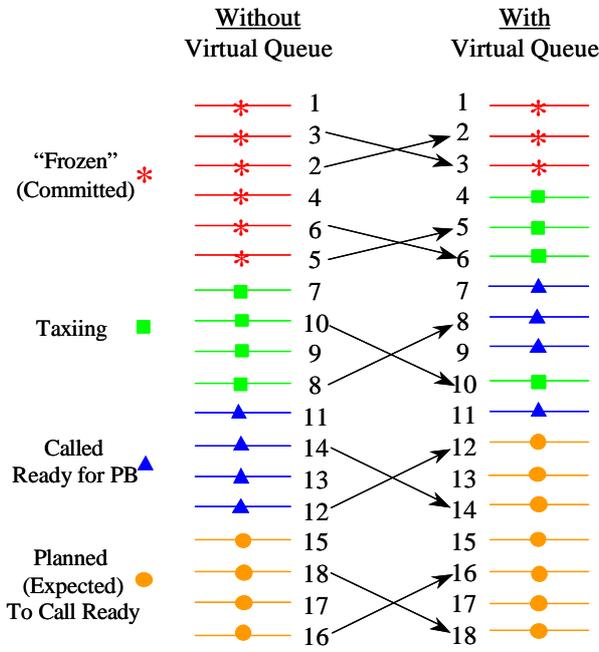


Figure B.5: Managing the departure sequence of the same 18 flights, with and without the implementation of a Virtual Queue (right hand side indicates the current optimal sequence)

Each line in Figure B.5 corresponds to a departing flight scheduled to take off within the time span that the virtual queue covers. In each case, the state of the departing aircraft is represented by a specific symbol as follows:

- a) “Frozen” (Committed to a particular runway)
- b) Taxiing
- c) Waiting for pushback clearance from the tower, after having Called Ready for Pushback (PB)

⁴⁸ ROP objectives are also presented in section C.1.1.1 of Appendix C.

and

d) Planned (Expected) to Call Ready (for pushback) within a pre-determined time horizon.

Due to high workload, it is very hard in most cases for air traffic controllers to mentally determine the appropriate timing and sequence of departures, while at the same time keeping in mind all constraints and satisfying all system objectives. The existence of the virtual queue may assist controllers to determine possible "aircraft takeoff swaps" within the same state or even between different aircraft states (arrows in Figure B.5) in order to optimize departure operations. The virtual queue may point out some of the optimal sequences that the controllers may not realize under heavy workload (see examples in Figure A.3 and Figure A.4). In addition, the virtual queue may be used to convert taxi delays to gate delays, which are less costly both for the airlines and the environment. That way, operational flexibility for the airlines can be increased without sacrificing fairness.

The VQM can be designed to exercise two levels of control on the allocation of runway time to different operations:

- Simple control of the size and / or sequence of takeoff queues and
- Time-based control.

In the second case it is expected that the solution quality will be enhanced, but at the same time the computational complexity of the problem may be increased.

Using the same shape coding as in Figure B.5, an example is presented in the following figures,

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

which describes the envisioned implementation of the Runway Operations Planning process as this is based on the virtual queue concept that was just described.

Runway Operations Planning can be performed at specific pre-determined time intervals or on a need basis. For example, even if there are pre-determined planning time points, in case a change in the existing airport surface conditions necessitates a new runway operations plan, planning can be performed even if the pre-determined planning time may have not come yet. So, at planning time T_i (Figure B.6), a specific runway operations plan (labeled “Runway Plan i-1”) has been generated since the previous planning time (T_{i-1} , not shown in the figure). At time T_i , certain aircraft from the generated runway operations plan (four (4) aircraft denoted by “*” in Figure B.6) are considered “frozen” and therefore, they cannot be included in the next planning session. The remaining aircraft are included in the next planning window, which also contains aircraft that are expected to call ready for pushback but in reality they have not entered the system yet (e.g. aircraft 8). In total, the new planning window contains eight (8) aircraft (numbers 1 through 8 in Figure B.6). So, at planning time T_i , “Runway Plan i” is generated for aircraft 1 through 8.

Subsequently, a few more aircraft from the pool of available departures are considered (aircraft 9 through 15 in Figure B.6) but not all of them are sure to be included in the next planning window. When the next planning time T_{i+1} arrives, the first three aircraft from “Runway Plan i” are “frozen” (Figure B.7). The new planning window in this case contains the remaining aircraft from the previous runway operations plan (aircraft 4 through 8), as well as all those aircraft that have actually called ready for pushback (aircraft 9 through 12 in Figure B.7). Contrary to the case in Figure B.6, no aircraft “expected to call ready for pushback” are included in the next

planning window in this case. Hence, at planning time T_{i+1} , the new runway operations plan (labeled “Runway Plan $i+1$ ”) is generated for aircraft 4 through 12 in Figure B.7.

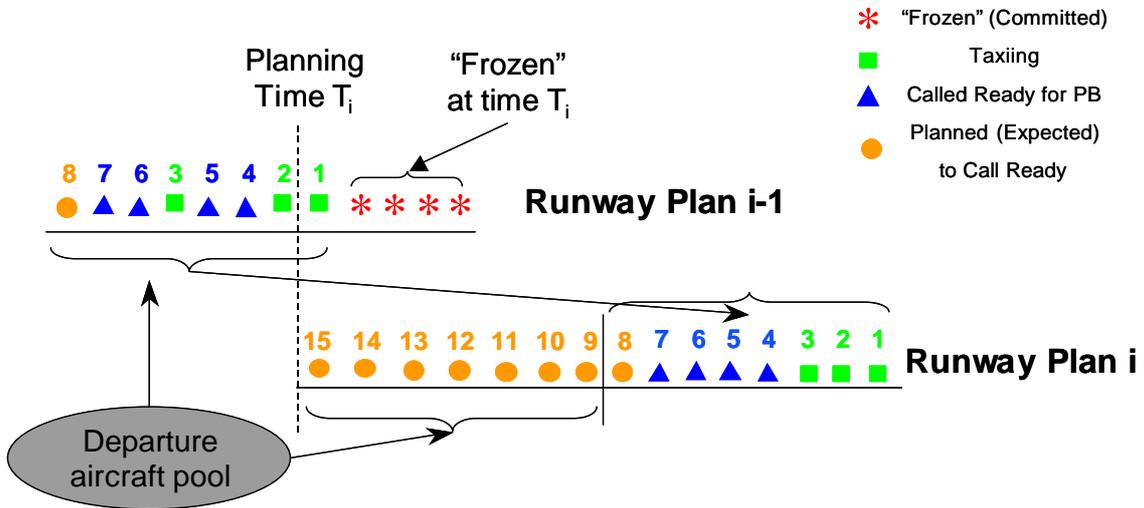


Figure B.6: ROP implementation – “Expected” aircraft ARE included in the planning

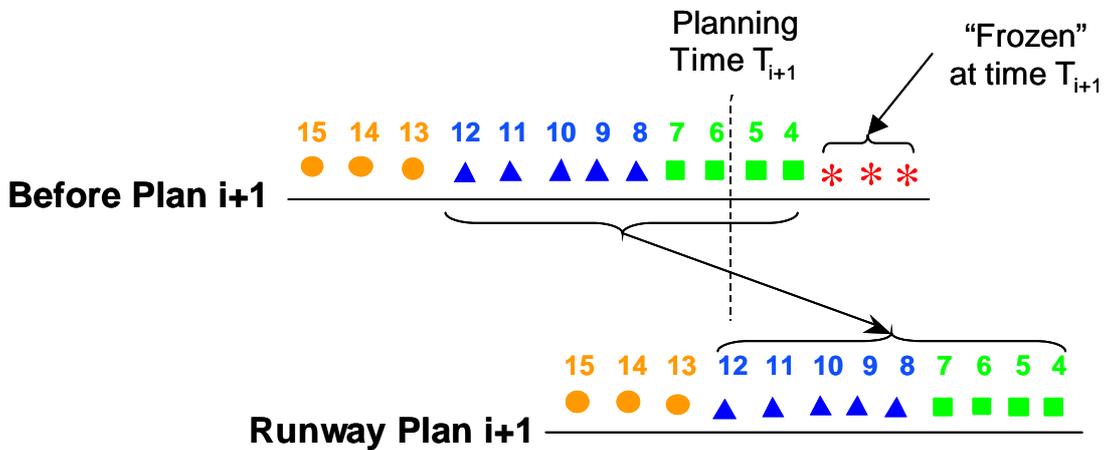


Figure B.7: ROP implementation – “Expected” aircraft ARE NOT included in the planning

B.3 The Gate Manager

The Gate Manager is the Departure Planner component that assists the controllers in determining the pushback schedule, subject to the uncertainty associated with airline gate operations. Initial runway assignments for departing flights may also be an important part of the Gate Manager’s task.

Functionality

Being the first Departure Planner component that can have an effect along the departure flow, the Gate Manager incorporates and processes data generated from the rest of the Departure Planner system components, as depicted in the “free body diagram”, shown in Figure B.8.

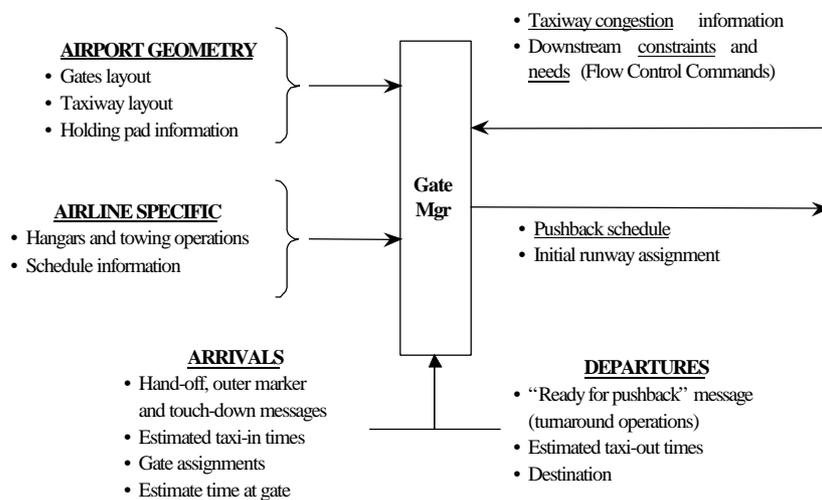


Figure B.8: The Gate Manager

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

Note that, arrows pointing inwards to the Gate Manager carry information (flight status data, system constraints) coming from other elements, which are adjacent to the Gate Manager in the system architecture, or from other National Airspace databases that exchange data with the Departure Planner (e.g. Surface Movement Advisor (SMA), Center TRACON Automation System (CTAS)). On the other hand, arrows pointing outwards from the Departure Planner component convey to the rest of the system commands and requests generated by the Gate Manager function. A similar convention is used to read the “free body diagrams” presented for the remaining system components that are described in the following sections.

Initially, based on traffic information from the gates, the ramp area and the taxiway system (Figure B.8, top left and top right data blocks), the Gate Manager assesses the current airport situation and suggests a feasible pushback schedule within a pre-determined planning horizon. At this point, data necessary for an accurate estimate of the current and projected demand (possibly obtained from the SMA database) are airline specific data, such as hangar status, current towing operations, and flight specific data local to each gate, such as destination, “turn-around readiness” messages, and taxi-out time estimates (Figure B.8 middle left and bottom data blocks). In addition, downstream constraints such as gate holds and Ground Delay Programs (Figure B.8, top right data block), which usually involve many cancellations, delays and gate rescheduling must be communicated to the Gate Manager as soon as the related information is available from the FAA central flow control (System Command Center).

Many of the airline operations, especially the ones performed before aircraft are actually ready to push back from their gates, are not observable by the controllers. For example, oftentimes

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

aircraft will call ready for pushback before their gate operations are actually complete, anticipating a delay between the call for pushback and the actual time that a clearance is granted. Sometimes, delays and cancellations due to inclement weather or mechanical problems, result in aircraft being held at their gates and cause unexpected gate blockages. With the Departure Planner, such situations can be observed by controllers through the exchange of pertinent information between the Gate Manager and the Database Management System.

Supporting Analysis

One control strategy is to limit the number of aircraft released in the system by holding some of them at their gates. This method is fairly easy to implement, however, it may raise gate capacity issues, since it transfers runway queue delays to gate delays [73]. Controlling the gate release times with the help of the Gate Manager provides the controllers with a unique opportunity to implement the above control strategy and control the size of takeoff queues and the sequencing of aircraft within the queues.

In view of this control option, Airline Service Quality Performance (ASQP) data was studied in [97] and a simple dynamic queuing model for departure operations was developed and used to analyze airport surface operations. Based on the analysis results, simple departure control strategies were suggested for the purpose of alleviating surface congestion. As presented in [97], an important parameter in the problem of departure control appears to be the number N of aircraft that are present in the system while an aircraft is taxiing out from its gate to the assigned departure runway. It was realized that, as the number N increases, the expected value and

variance of the taxi out time also increases, as it is also shown in Figure B.9 ([35]) for one year of data from Boston’s Logan airport.

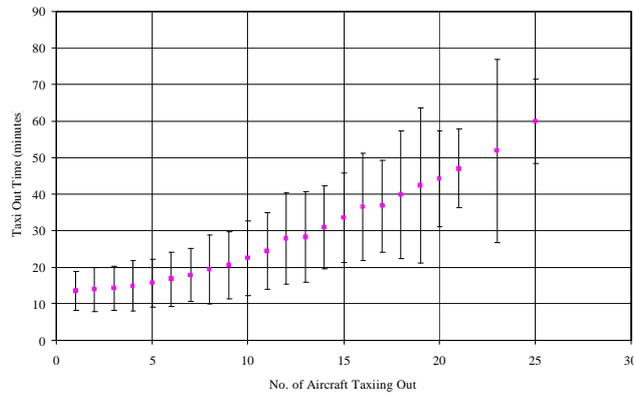


Figure B.9: Taxi out time as a function of airport congestion (BOS: 1997/09 – 1998/08)

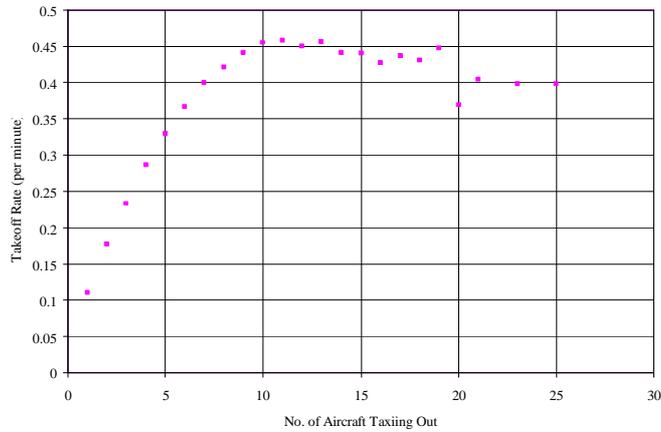


Figure B.10: Takeoff Rate saturation point (BOS: 1997/09 – 1998/08)

In addition, for each airport and under any configuration, there is a runway saturation point beyond which there is no significant gain in takeoff rate even if controllers keep releasing more aircraft in the system, as shown in Figure B.10. It is observed that, as controllers release more

aircraft in the system, less than the released number of aircraft actually takes off and the remaining form congestion queues on the airport surface (Figure B.9 and Figure B.10 from [35]).

Based on downstream requests the schedule can be adjusted through gate release control to feed the takeoff buffers with the requested number of aircraft. The system-wide objective of maximizing airport throughput is addressed and pre-allocated departure slots can be met. Engine-running times are also minimized and compliance with environmental emissions regulations is achieved, while gate-blocking delays are significantly reduced. Furthermore, the airlines benefit from fuel savings and late passenger / baggage accommodation by remaining at the gate until they can actually be accepted in the taxiway system, as opposed to pushing back on time and being delayed in holding pad areas or in taxiway and runway queues.

B.4 The Taxiway Entry Manager

The Taxiway Entry Manager determines the sequence and timing of release from the ramp into the taxiway system for aircraft that have pushed back from their gates and entered the “ramp buffer” in Figure B.1. It considers system objectives related to the total time that each aircraft spends on the ramp or taxiing with its engines running.

Functionality

The Taxiway Entry Manager can affect departure operations by regulating the flow of aircraft through the interface between the gates and the taxiway system, which was identified as another

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

possible control point in the departure flow [72], [73]. The Taxiway Entry Manager provides a means of indirectly controlling the runway takeoff queues, by controlling the total number of departing aircraft that will be distributed to them. Note that, depending on the specific airport geometry and complexity, this interface can take various forms. At Logan, there is a set of entry points to the taxiway system with little or no ramp area around the various terminals, while other airports, such as Atlanta Hartsfield or Chicago O'Hare, have a ramp area of considerable size adjacent to the terminals.

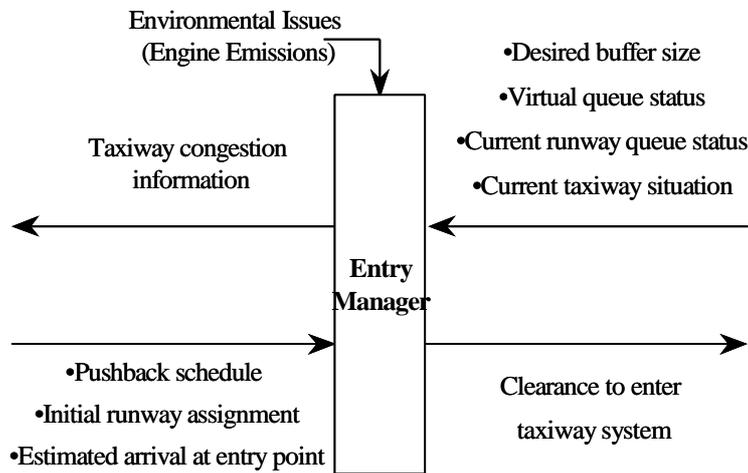


Figure B.11: The Taxiway Entry Manager

The current and projected taxiway situation (congestion levels) feeding back from simple observations or from sophisticated airport surveillance systems (in the future) and the takeoff queue (buffer) size feeding back from the Mix Manager are the most critical pieces of information for the Taxiway Entry Manager (Figure B.11, top right data block). Accurate short-term estimation of pushback operations and prediction of the demand to enter the taxiway system must also be performed and the results fed into the Entry Manager, in order to avoid overloading

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

the entry points (Figure B.11, bottom left data block). All the above information is processed under the constraints of environmental regulations on aircraft engine emissions (Figure B.11, top input). The outcome of this system element (Figure B.11, bottom right data block) could be a feasible schedule of release times for aircraft to enter the taxiway system, which also meets the system objective of minimizing aircraft taxi times and therefore engine-running times, emissions and airline direct operating costs.

“Engine-start” time control is an additional issue pertaining to the environmental impact from aircraft engine noise and emissions, which deserves further examination. In current operations, only pushback and taxiway entry clearances are commanded by terminal ATC and the exact time that aircraft engines are started is left entirely to the pilot’s discretion. The Gate and Entry Managers could possibly schedule the movement of aircraft under the additional objective of postponing engine start times until as close to the taxiway entry clearances as possible. This is an example of the kind of coordination that is necessary between two tactical subcomponents, the Gate Manager and the Taxiway Entry Manager. At Logan, where ramp space is limited, the two subcomponents can be merged into a single tool.

Taxi path modifications – Runway reassignments

It should be noted that departing flights push back from their gate with an initial runway assignment, which they usually maintain until takeoff. Nevertheless, initial runway assignments and taxi paths are not always constant. When for example, a taxiway segment is unexpectedly blocked or there is a short-term or scheduled runway configuration change or the load in a

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

certain takeoff queue is high, a runway reassignment may be needed. Had the controllers had prior knowledge of the particular circumstances that led to the need for a runway reassignment, their pushback and taxiway entry clearance deliveries could have been different and also additional workload imposed on them and additional cost and delays imposed on the airlines could have been prevented. In the context of the Departure Planner, reassignment and rerouting may also be necessary in order to implement the Virtual Queue planned sequence, which is very dynamic.

Potentially, an additional subcomponent could be inserted at this point in the system architecture to assist controllers with their runway reassignment and taxi routing decisions and to help them implement the Virtual Queue. Such a component is not currently included in the proposed Departure Planner architecture because it is believed that the taxi ground operations are not amenable to automation and are better handled by the experienced air traffic controllers. In the current operational environment, operations at the gates / ramp and at the runway, are supported by automation systems, such as CTAS and SMA, which are already in place and can be interfaced to the proposed Departure Planner components. On the contrary, automation during taxiing is hindered by the lack of surveillance information. In fact, in managing taxi operations, air traffic controllers obtain (visually) downstream information regarding the size and sequence of each takeoff queue, as well as the current status of each of the available runway queues. Such information can possibly come from downstream Departure Planner components such as the Mix Manager (see next, section B.5).

B.5 The Mix Manager

The Mix Manager regulates the release of departing aircraft from each “runway buffer” onto the corresponding runway (runway A or B in Figure B.1) as well as controls the release of aircraft from the runway crossing queues building up on the taxiway segments. The coordination of operations on dependent runways and the mixing of arrivals, departures and crossing operations on a single runway are its main tasks. This component of the architecture is the one that can benefit primarily from the core Runway Operations Planning research work presented in this document.

Functionality

Air traffic controllers usually prefer to assign arrivals and departures to different runways. However, this is not always feasible, especially in tightly constrained airports such as Boston Logan. In many runway configurations, the runway resource utilized by departing aircraft is shared with arriving aircraft, which in most cases have priority over departures. In addition, the runway system is frequently shared with taxiing aircraft that have to cross active runways. As illustrated in Figure B.12, the controllers often have to introduce gaps in the arrival stream in an effort to accommodate departures between arrivals and to allow taxiing aircraft to cross active runways. Sharing of the runway resources introduces a strong coupling between the arrival and the departure streams [71], [72], [73]. This suggests that we must consider and manage airport runway resources as sets of dependent runways, as opposed to individual runways.

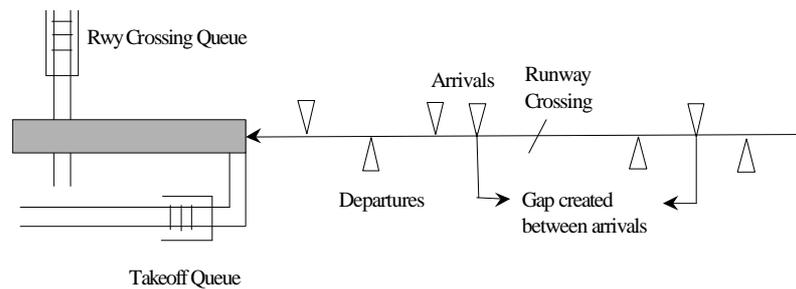


Figure B.12: Runway Operations Schedule: Mixing of Arrivals, Departure and Runway Crossings
(source: [74])

The interaction of the Mix Manager with the rest of the aircraft flow at an airport system is described in Figure B.13. As suggested it is the connective component between terminal airspace traffic (departures ascending within the terminal airspace and arrival flow approaching the airport) and airport surface traffic (the set of departing or arriving aircraft that are physically present on the taxiway system).

As shown in Figure B.13, working under a given runway configuration and a specific mode of operations (Figure B.13, top right data block), the Mix Manager processes the following inputs:

- Projected takeoff demand information, based on inputs from the actual and projected pushback schedule (Figure B.13, bottom left data block).
- Projected landing demand information from the final approach arrival queues that are forming in the terminal area (Figure B.13, middle right data block) and
- Data on downstream constraints, such as Miles (Minutes) In Trail and departure fix capacities (Figure B.13, middle right data block).

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

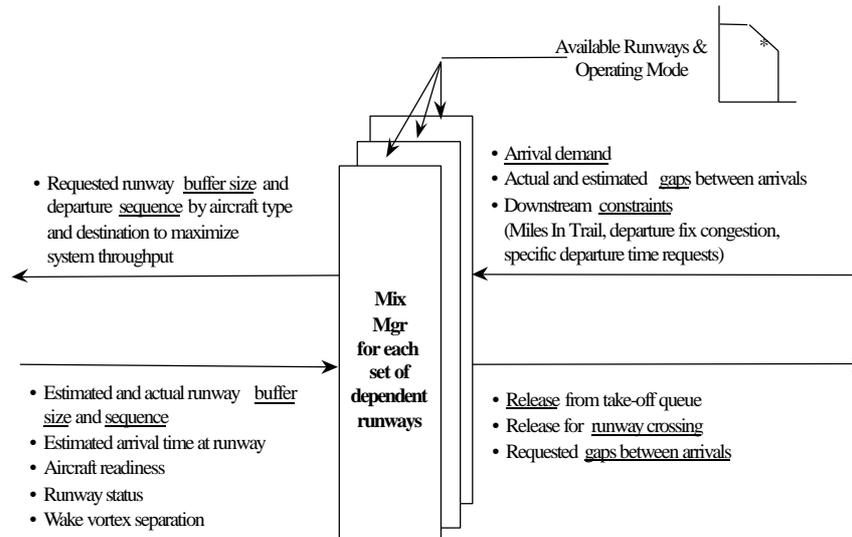


Figure B.13: The Mix Manager

Collaborative Decision-Making (CDM) can play an important role at this point, in providing accurately updated demand information (cancellations and delays) to the air traffic controllers and to the mixing function of the Mix Manager.

The main output generated from the Mix Manager is the suggested schedule of aircraft release from the takeoff and runway crossing queues (Figure B.13, bottom right data block). Requests for gaps in the arrival flow could be given to the TRACON controllers, in order to implement the suggested takeoff releases. In addition, specific tactical suggestions on the sequence and size of takeoff queues can be communicated to the tower controllers as a basis for carrying out efficiently the gate pushback and taxiway entry processes.

Supporting Analysis

In Logan configuration 22L / 22R / 27, which was presented in Figure A.1, arrivals using runways 27 and 22L have to cross runway 22R to reach the terminal area. Crossing aircraft queue in the taxiway segments between runways 22R and 22L but when there is no more taxiway space for queuing aircraft, the departure stream on 22R has to be interrupted for crossings to occur and for making runways 22R and 27 available for further landings.

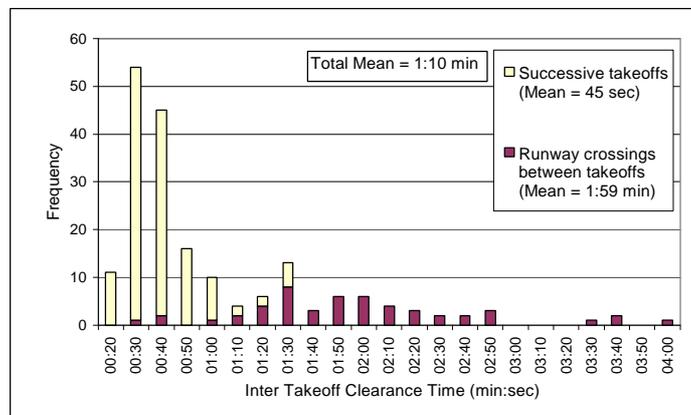


Figure B.14: Runway Crossing Effect on Inter-Takeoff Clearance Times (Graph source: [74], Data source: Local Controller Clearances, Runway 22R, Logan Airport, 12-2-1998, 4-9PM)

Figure B.14 clearly demonstrates how takeoff operations on runway 22R are affected by the presence of crossing aircraft. When one or more crossings were operated between successive takeoffs on 22R, the mean time between the two takeoff clearances for the data sample presented is 1 min and 59 seconds. The mean value is reduced when successive operations occur with no crossings interjected between them. The overall mean time for the whole data set is 1 min and 10 sec.

Appendix B: Conceptual Architecture of a Departure Planning Decision-Aiding System

The Departure Planner cannot be developed independently from CTAS or other arrival automation tools, which carry information critical to Departure Planner for successful configuration planning and arrival/departure mixing. In fact, the arrival-departure interaction introduces a new complex challenge for existing tools, such as CTAS, which will now have to be enhanced to take departures into account. The Departure Planner system can have important inputs to CTAS and especially Active FAST (Final Approach Spacing Tool) [84], such as the runway crossing and takeoff queue information. These inputs can then be used to determine the most appropriate sequencing and tactical spacing of arrivals (introducing the necessary gaps in the arrival stream, Figure B.12).

Appendix C

RUNWAY OPERATIONS PLANNING- FUNCTIONAL AND RELATIONAL DECOMPOSITION ANALYSIS

Functional and Relational Decomposition, as defined in section 3.3.2, are system analysis techniques commonly used in the design of electromechanical systems in order to understand the links between “what the system is required to do” (functional requirements) and “how the system will perform its functions” (design parameters). Given the decision to approach the design of a solution methodology for the Runway Operations Planning problem from a systemic point of view, these two techniques are applied here on the sets of functional requirements and design parameters of the complex Runway Operations Planning problem.

C.1 Functional Decomposition Analysis

C.1.1 ROP “Customer” Needs & Functional Requirements

An airport is by nature a multi-objective environment and therefore, the functional requirements of a decision-aiding system that is deployed to assist controllers in planning airport ground operations are largely dictated by the operational and financial interests of the stakeholders involved, such as the airport users (airlines, passengers), and the Air Traffic Management (ATM) service providers (airport authority, air traffic controllers). In addition, there are legitimate concerns of surrounding communities to reduce or at least to maintain at current levels the environmental impact from aircraft noise and emissions.

Each of the interested parties attaches different “weights” to system objectives. In some cases, interests can be mutually supporting. For example, the airlines’ effort to avoid unnecessary gate and taxi delays and hence reduce economic inefficiencies and unnecessary costs during the taxi process is aligned with the environmental objective of reducing noise and emissions. In addition, the passengers’ interest in having punctual and reliable transportation service is attended. In other cases, objectives may be more competing than supporting. For example, an envisioned and absolutely necessary airport throughput capacity enhancement, mandated by traffic demand forecasts, competes with environmental objectives. Nevertheless, the level of safety in air transportation should be at least maintained if not enhanced and new technical systems or procedures developed for ATM should not in any way increase air traffic controllers’ workload.

Appendix C: Runway Operations Planning - Functional and Relational Decomposition

Some of the main system goals in runway operations management are:

- To maintain safety by:
 - Complying with regulations and requirements such as wake vortex separations or Miles (Minutes) In Trail restrictions and
 - Maintaining controllers' workload at acceptable levels without, however, running the risk of reducing their vigilance and control of the traffic situation by automating many of their current tasks
- To enhance resource efficiency (gates, ramp, taxiways, runways) by:
 - Increasing or even maximizing system throughput for each available runway configuration by servicing requests for runway time as quickly as possible and balancing the load of operations between all active runways
 - Minimizing taxi times and
 - Minimizing pushback and other clearance delays
- To reduce environmental impact by:
 - Controlling aircraft engine emissions, to which the minimization of taxi times also contributes and
 - Considering noise regulations and constraints when determining airport configuration plans, for the benefits of communities surrounding the airport,
- To reduce economic inefficiencies by:
 - Minimizing "engine-run" times (related to taxi times) and therefore saving the airlines from unnecessary fuel burn and the associated costs and

Appendix C: Runway Operations Planning - Functional and Relational Decomposition

- Guaranteeing fair treatment for all airport users, such as different airlines or different flights of the same carrier
- Coordinating with the arrival stream in order to preserve arrival priorities and to minimize or preferably avoid airborne holding of aircraft, which also contributes to environmental benefits and

Note that, since the air traffic controllers have final authority in determining the criteria to be used for runway operations optimization at each point in time, optimization criteria must be selected in a way that has intuitive meaning to them. Such examples of criteria are the airport departure or total operations rate (throughput) or the average taxi times. Therefore, a “transformation” is necessary in order to translate the above goals to specific mathematical functions, which will be used to evaluate the system behavior with respect to one or more of the criteria in hand. For this purpose it is helpful to separate conceptually the goals in the list mentioned above into two layers: a) general system objectives and b) airport system objectives, as illustrated in Figure C.1. Satisfying the latter is a means to fulfill the former ones.

The relations indicated by arrows in Figure C.1 may be interpreted as a further specification of goals. In fact, this separation in different layers helped in specifying the various categories of “customer needs” (or attributes as mentioned in [116]) that we chose to address in this research for designing a solution algorithm for the Runway Operations Planning problem.

Appendix C: Runway Operations Planning - Functional and Relational Decomposition

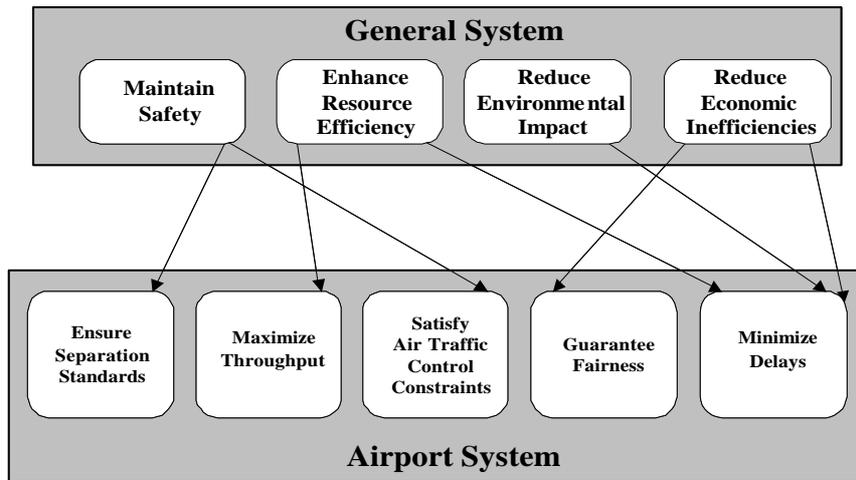


Figure C.1: Layers of “customer” needs / attributes in Runway Operations Management

These needs in the Customer Domain, as well as the functional requirements that represent those needs in the Functional Domain are shown in Table C.1. For example, maintaining separation standards is one of the necessary functional requirements that addresses the “customer” need for safety of runway operations, where “customer” in this case is the groups of users that utilize the airport system, i.e. airlines and passengers. Also, in a similar fashion, maximizing throughput is assumed to represent the need for efficient operations at an airport system and minimizing delays is the functional requirement that represent the need for mitigation of environmental impacts.

Domain Name	Attributes / Requirements				
Customer (Attributes)	Maintain Safety	Enhance Resource Efficiency	Satisfy ATC Restrictions	Consider Fairness Issues	Reduce Environmental Impact & User Inefficiencies
Functional (Functional Reqs)	Maintain Separation	Maximize Throughput	Satisfy Downstream Flow Constraints MIT, EDCT, DSP	Fairness Constrain Position Shifts	Minimize Delays

Table C.1: ROP “Customer” Needs and Functional Requirements

This categorization of functional requirements is the first step of the Runway Operations Planning problem *Functional Decomposition*, which separates the system functional requirements in subsets. In the next section, system parameters will be introduced and mapped to these requirements.

C.1.2 ROP Design Parameters & Process Variables

Having determined the elements of the ROP problem in the Customer and Functional Domains, the Physical and Process Domain remain to be populated in order to understand the mapping between different domains and investigate how it can be decomposed.

The Physical Domain contains those design parameters that specify the functional requirements and the Process Domain contains those variables that can mathematically represent the process of satisfying each functional requirement. For example, the need to maintain safe operations functionally requires maintaining appropriate separation between successive takeoffs. This is a requirement that depends only on one design parameter: the separation matrix and its entries used for determining the safe time separation (in seconds) between successive takeoffs on the same runway. Of course, the separation values depend on the weight class of the leading and trailing aircraft. The most common form of a separation matrix is shown in Table C.2. So, if there is a pair of successive takeoffs and the leading one is operated by a Heavy (H) aircraft, while the trailing one is operated by a Large (L) aircraft, safety is ensured only if the trailing Large is cleared to take off 120 seconds after the leading Heavy has cleared the runway.

Appendix C: Runway Operations Planning - Functional and Relational Decomposition

		Trailing Aircraft		
		H	L	S
Leading Aircraft	H	90	120	120
	L	60	60	60
	S	60	60	60

Table C.2: Separation criteria (in sec) for three weight classes (Heavy, Large and Small)

Finally, the actual problem variable, that determines mathematically whether safe separation is maintained or not is the length of the time slot between two successive takeoffs. A similar domain analysis for each of the main system requirements that this design aims to satisfy, are shown in the complete table of Table C.3.

Domain Name	Attributes / Requirement / Parameter / Variable				
Customer (Attributes)	Maintain Safety	Enhance Resource Efficiency	Satisfy ATC Restrictions	Consider Fairness Issues	Reduce Environmental Impact & User Inefficiencies
Functional (Functional Reqs)	Maintain Separation	Maximize Throughput	Downstream flow constraints MIT, EDCT, DSP	Fairness Constrain Position Shifts	Minimize Delays
Physical (Design Params)	Separation Matrix (const)	Weight Class/ Traffic Mix	Aircraft to Slot assignments	MPS Value	Individual Aircraft Taxi-Out Delays
Process (Process Vars)	Class Slot Length	Sequence of Class Slots	Aircraft Takeoff Positions	MPS Variable (const)	Sum of Individual Delays

Table C.3: The Complete Set of ROP Design Domains

Elements in this table, including those that have not been introduced yet, such as the MPS (Maximum Position Shifting) concept, will be explained in later sections of this appendix. Focusing on the runway resource, we assume “maximizing throughput” to be the only functional requirement assigned to satisfying the need for “enhancing resource efficiency”, “weight class / traffic mix” to be the corresponding design parameter and “sequence of weight class slots” to be the problem variable that mathematically determines the actual value of runway throughput. The

“delay minimization” requirement is directly associated with satisfying the system need of “reducing environmental impact and system inefficiencies”. The rest of Table C.3 can be explained in a similar fashion.

C.2 ROP Problem Objectives & Constraints

The full description of the Runway Operations Planning problem involves the formulation of the problem objectives to be met and the constraints to be satisfied by the problem solution. Depending on the chosen mathematical formulation and on the nature of each individual functional requirement, some of the functional requirements in Table C.3 can be treated either as objectives or as constraints. *Throughput maximization* and *minimization of taxi delays* are chosen as the problem objectives, because, in real world operations, these are the most common performance concerns of air traffic controllers, traffic flow managers, airlines and passengers. As far as the problem constraints are concerned, the most important constraints that commonly impact runway operations are outlined in the following sections. There are two possible ways to structure these constraints: according to the severity of their impact on operations or according to their source.

C.2.1 Constraint Types

Based on the level of impact that a constraint violation may have on the airport and the ATC system, two constraint types can be distinguished:

- Hard constraints, which are inviolable and must be satisfied by all generated solutions and

- Soft constraints, which can be violated but the less the violation the better the quality of the solution is.

An example of a hard constraint is the wake vortex restrictions between successive runway operations. Takeoff clearance between successive departures must be timed in a way to allow for wake vortices generated by the leading aircraft to dissipate without jeopardizing the safety of the trailing aircraft. The system must be designed so that, irrespective of the mathematical formulation of the objective function, no wake vortex restrictions are violated in the resulting plan. Thus, departure plans that violate hard constraints will be ruled out of the solution.

A soft constraint allows more planning flexibility to the optimizer but still deteriorates the objective function when violated. For example, a flight is optimally planned to take off within a certain time window. If that window needs to be even slightly exceeded (violated) by the aircraft, the resulting takeoff plan is still feasible but not optimal any more. The degree of the slot violation determines how far from optimality the system (objective function value) will be driven.

Soft constraints can also be addressed and formulated as objectives, either separately from the two main performance objectives mentioned above (throughput maximization and taxi delay minimization) or even incorporated in one of them (e.g. takeoff time window constraints can be merged into the taxi delay minimization objective). Obviously, when designing system constraints for implementing a ROP system, it is wiser to use soft constraints when possible.

That is why, in the problem formulation presented in Chapter 3, most functional requirements were treated within the constraint set of the formulation, in order to keep the objective functions as simple as possible, even though some could have been treated as objectives. It is important to mention that, when planners need to establish hard constraints, they must take extra care not to over-constrain the system and run the risk to drive it to infeasibility.

C.2.2 Constraint Categories

Based on the constraint source, the following categories of constraints can be identified:

- *General operational constraints*, which are imposed by the ATC operators in order to handle high volumes of traffic in a safe manner. A category of such constraints is the wake vortex separation restrictions at the takeoff point. Another example is the Miles In Trail or departure fix restrictions imposed on airborne aircraft, which are primarily due to en route traffic congestion or to the Standard Instrument Departure (SID) structure around the airport.
- *Special operational constraints*, which are introduced by individual controller or pilot requests and are specific to the way runway operations are carried out at a particular airport. A representative example of specific operational constraints is the controller inputs that often introduce new restrictions and requirements (e.g. Expected Departure Clearance Time (EDCT) slots, limiting pushback to takeoff swaps, SID separations) or modify existing procedures. Furthermore, pilot or airline inputs can be considered as specific operational constraints, when a pilot requests to use a specific runway that is not active in the current configuration (this can happen for example when an aircraft is too heavy and needs a longer takeoff runway), or when an airline requests priority handling for specific flights or last

minute schedule adjustments to accommodate connecting passengers and

- *Physical constraints*, which exist in the system due to the airport topology and runway and taxiway location relative to the terminals and gates. These are constraints that primarily affect aircraft taxi times and runway crossings. For example, if at a given airport it takes at least 10 minutes for an aircraft to taxi at regular taxi speed from its gate to the assigned runway, any generated plan that requires this operation to occur in less than 10 minutes is unacceptable.

All types of restrictions and system constraints have to be mathematically formulated in order to be easily included in the implementation of an ROP optimizer. In this case, constraints can be formulated as limitations imposed on the position number that an aircraft can take in the takeoff queue. This approach is chosen primarily because, in real-world operations, air traffic controllers usually plan surface operations in terms of sequencing aircraft positions as opposed to commanding the exact timing of each aircraft operation. Therefore, based on their individual experience and skill, air traffic controllers translate the various timing and spacing constraints to pushback, taxi or takeoff position constraints. Some examples of mathematical representations of constraints follow:

- a) *Wake Vortex separations*: there have been studies [96] in the past, in which it was realized that, in the case of arrivals, it is optimal in terms of landing throughput to “group” and clear to land aircraft of the same weight category together. Similar results can be easily shown in the case of departures also. Instead of consecutively releasing aircraft from different weight categories, controllers can achieve a higher departure throughput by sequencing the aircraft in such a way that aircraft of the same weight class take off together. For example, let us

Appendix C: Runway Operations Planning - Functional and Relational Decomposition

consider only three weight classes: Heavy (H), Large (L) and Small (S). The time separation criteria (in seconds) are shown in Table C.2. Based on those criteria, the sequence of classes LLLHHSSS can be released through the same runway in a total of 510 seconds, while the sequence of classes LHLHLSSS needs at least 540 seconds to be handled. Therefore, one way to ensure that departing aircraft are “grouped” is to limit the absolute difference of sequence positions that any two aircraft of the same weight class can have. If i and j are two aircraft in the same weight class and P_i, P_j are their respective takeoff sequence positions, then it must be:

$$|P_i - P_j| < k, \forall i, j \quad (C.1)$$

where k is the number of aircraft of that weight class within the group of aircraft that the optimizer currently handles. So, if there are 4 Heavy aircraft currently in the optimizer, a generated plan that requires any two of those aircraft to be more than 3 positions apart in the takeoff sequence is not acceptable. Note that, depending on the weight class composition of the group of departures that is being planned, a “grouping” constraint such as the one above, must be used with caution. For example, if there are no heavy aircraft involved in the group, then, based on the separation requirements in Table C.2, “grouping” takeoffs of the same weight class will not improve the final runway throughput achieved. In such a case, a hard inviolable “grouping” constraint may over-constrain the problem and make it difficult to reach to a feasible solution.

- b) *Operational constraints* that impose aircraft separations such as Miles In Trail are always stated in terms of airspace separation distance, which must be translated to relative position constraints. Therefore, it must be:

$$|P_i - P_j| \geq \Delta P \quad (C.2)$$

where P is the takeoff sequence position and ΔP is the difference between two positions. This signifies that aircraft i and j , which are subject to Miles In Trail restrictions can only take off at least ΔP positions apart in order to ensure that the In Trail separation is achieved when they will be airborne.

- c) *Other types of operational constraints* may restrict the time point or time slot that an aircraft can be released for takeoff. In that case, time can be translated to specific takeoff position constraints. For example, an EDCT restriction can be formulated either as a specific position in the takeoff sequence:

$$P_i = P_1 \quad (C.3)$$

or as a range of acceptable positions:

$$P_1 \leq P_i \leq P_2 \quad (C.4.)$$

Special flights that have urgent takeoff priority, such as lifeguard / medical flights, must take off ahead of most or all flights and the formulation in this case is slightly different:

$$P_i = 1 \text{ or } 1 \leq P_i \leq P_3 \quad (C.5).$$

Any other type of priorities can be modeled as an inequality constraint:

$$P_i < P_j \quad (C.6).$$

- d) One possible way to maintain *fairness* among airport users is to limit the level of deviation from a “First Come First Serve” policy. This can be achieved if the controllers use a “Maximum (Takeoff) Position Shifting” policy. The way to formulate this as a constraint in

the optimizer is:

$$|P_{\text{PB}} - P_{\text{TO}}| \leq \Delta P_{\text{limit}} \quad (\text{C.7})$$

where ΔP_{limit} is the maximum number of sequence position shifts that a flight can tolerate between pushback (PB) and actual takeoff (TO).

After the functional decomposition, which defined the five different categories (columns) shown in Table C.3, Relational Decomposition is the next step that helped in the design of a solution methodology for the Runway Operations Planning problem. Relational decomposition is performed in order to analyze the ROP problem properties and determine problem functional levels that are independent or loosely coupled, with an objective to write a Design Matrix equation (such as the one in Figure 3.4) for the system. This process leads to systems that offer a higher chance of robust behavior, because any failure (i.e. in the case of ROP, any operational event that mandates a plan change) affects only a certain level of the system and does not propagate to the rest of the independent levels.

C.3 Relational Decomposition Analysis

It has already been determined that the two main system performance functions to be optimized, are the throughput and the system delays (system functional requirements in the second row of Table C.3). The remaining functional requirements, such as maintaining system safety and fairness among users or satisfying downstream flow restrictions, will be included in the set of constraints of the ROP problem formulation.

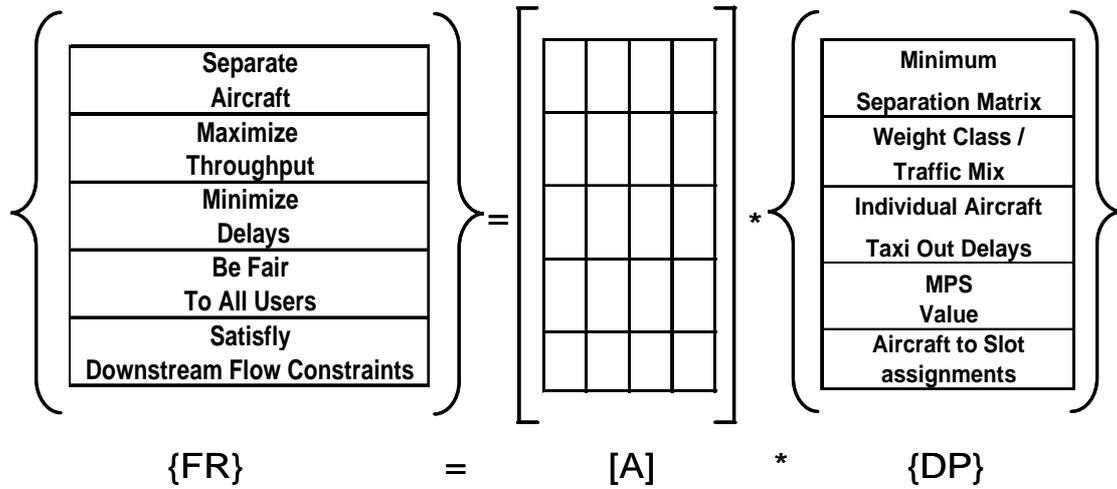


Figure C.2: ROP Design Matrix Equation – Step 1

The problem Design Parameters {DP} that will be included in the problem formulation are the elements of the third row of Table C.3. The relational decomposition analysis helps in determining which of those parameters are linked to which functional requirement(s) {FR} and in filling the entries of the design matrix [A] in Figure C.2. Let us now examine each of the functional requirements separately:

C.3.1 Separate Aircraft (Safety)

Ensuring operational safety by maintaining the mandated separation standards between successive operations is a requirement that must be satisfied regardless of the value of any of the input or output design parameters. The only input parameter that affects this requirement is the separation matrix, which has pre-assigned entries that remain constant. The (input or final output) values of:

- The traffic mix in terms of weight class, i.e. the percentage of departure demand by aircraft of Heavy, Large and Small weight class category
- The Maximum Position Shift (MPS) parameter
- The final aircraft takeoff sequence assignments and
- The individual delays that each operation will have to absorb as a result of the problem solution (runway operations plan)

cannot change the fact that separation between successive runway operations must be maintained. Since the pre-assigned entries of the separation matrix remain constant, the functional requirement of maintaining safety and the separation matrix design parameter will be omitted from the design matrix equation for the remainder of the document without loss of generality.

C.3.2 Maximize Throughput

The final runway departure throughput is a result of the specific runway operations plan that is given as a solution. However, assuming that all available runway time is utilized and there are no “runway-idle” gaps in the runway operations schedule, the runway throughput (as a mathematical function) is determined by how short or long is the time separation between successive operations. The latter depends on the weight class of the aircraft that participate in the leading and trailing operation of each pair of successive operations. So, if we assume that the departure sequence has been decided in terms of weight classes, throughput is determined. At that point, provided that an aircraft of the required weight class is used for each takeoff position in the sequence, the departure throughput value cannot be affected by which particular aircraft

fills each sequence position. Therefore, departure runway throughput can be assumed not to be directly associated with aircraft-specific design parameters, such as the fairness-related Maximum Position Shift (MPS) value, the “aircraft to slot” assignments and the resulting taxi-out delays of the individual aircraft.

It can be argued that it is not a safe assumption because, even if a “throughput-optimal” weight class sequence has been determined, it is still possible that the latter may become infeasible due to downstream flow restrictions or other temporary special Air Traffic Control restrictions, which typically mandate that certain aircraft be assigned to certain takeoff sequence positions (“aircraft to slot assignments” design parameter). If these mandates are incompatible with the “throughput – optimal” weight class sequence, then either the latter needs to be abandoned and, a new sequence has to be devised, or “runway-idle” gaps have to be introduced in the takeoff plan, both of which yield a sub-optimal runway throughput value. How often this can happen, naturally depends on how frequently (or infrequently) Air Traffic Control restrictions are active.

For now, *it is assumed that design parameters other than the weight class mix are (at most) “weakly linked” to the functional requirement of throughput maximization.* This assumption is can also be supported by the following two arguments:

- a) *Schedule homogeneity*: The weight class mix profile of the schedules operated at most major airports is generally homogeneous, especially since the advent of regional jet aircraft in the National Airspace System, and the majority of the operated aircraft belong to the Large weight class category, with only a few Heavy and Small remaining. This means that, even if

downstream restrictions mandate specific takeoff sequence positions for certain aircraft, it is likely that these mandates will be satisfied just by swapping takeoff positions between two or more aircraft of the same weight class, without the need to abandon an already generated weight class sequence.

- b) *Frequency of occurrence* of Air Traffic Control restrictions: Not every aircraft is affected at all times by a downstream flow restriction. In fact, Idris [74] analyzed a month-long set of Boston Logan operational data (July 1998) in which, out of a total of 7,766 aircraft observed, almost 89 % (6896 observations) were not affected by any form of restriction⁴⁹.

Based on the above arguments and if a capital X symbol is used for “strong” links and a lowercase x for “weak” links, the design matrix equation in Figure C.2 so far assumes the form:

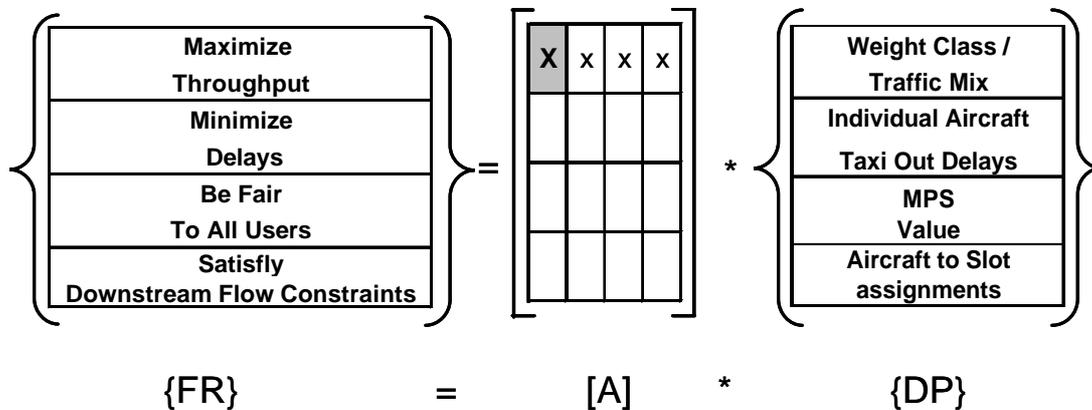


Figure C.3: ROP Design Matrix Equation – Step 2

⁴⁹ This however, could be attributed to the fact that the data came from a summer month with prevailing good weather conditions.

C.3.3 Minimize Delay

The second main system objective of minimizing departure aircraft delays is strongly linked to all the input or output design parameters that affect or are affected by the amount of time that each individual aircraft spends in the airport system, i.e. from gate pushback to takeoff release at the departure runway. These design parameters considered in the design equation, are:

- The individual delays for each departure operation (output)
- Maximum Position Shift (MPS) (input) and
- Final aircraft to slot assignments (output)

A “weak link” argument, similar to the one for throughput maximization, can be made for the relation between the functional requirement of minimizing delays and the weight class / traffic mix design parameter. The pre-arranged class sequence does play a role in the position of each aircraft in the departure sequence and therefore affects the taxi-out delay that each aircraft will absorb, especially if there exists only one aircraft of a particular weight class in the pool of available aircraft. If it is possible that a different class sequence will facilitate lower individual delays for specific flights, the decided class sequence could be altered, but it is also possible that the new weight class sequence will not yield the optimal throughput value that the previous one had and will not yield lower cumulative taxi out delays for all involved aircraft. So, assuming a fixed weight class sequence, the aircraft delays are only related to the three design parameters mentioned above.

C.3.4 Be Fair to All Users

The system requirement of treating all users equally may be enforced through the use of the Maximum Position Shifting (MPS) constraint. The optional use of the MPS parameter limits the amount of sequence position shifts that each aircraft is allowed to undergo between pushback and takeoff. For example, if $MPS = 2$, the 4th aircraft that calls the Air Traffic Control tower ready for pushback can neither be favored with an advancement to the first takeoff position (because $4 - 1 = 3 > 2$) nor can be delayed beyond the 7th takeoff position ($7 - 3 > 2$). The fairness functional requirement is strongly linked to those input or output design parameters that can impact the feasibility of the MPS constraint in the problem formulation:

- Maximum Position Shift (MPS) value (input) and
- Final aircraft to slot assignments (output)

or are affected by the existence of the MPS constraint in the problem:

- The individual delays for each departure operation (output).

When all MPS problem constraints are satisfied, certain aircraft delays follow as a result, but each aircraft has to settle with the delay assigned to it. This proposes that the fairness constraints are linked to the output design parameter of individual delays for each operation. It is unlikely that an aircraft in the plan will have high delays (above certain tolerable limits) because that is the main reason why the fairness (MPS) constraints were introduced in the first place, i.e. to prevent individual aircraft from suffering excessively high delays. If, however, this occurs, then air traffic controllers may choose to alter the final departure schedule in a way that may not guarantee fairness among users any more, because a controller's initiative for reducing the delay

of a specific flight may introduce takeoff sequence position changes that violate MPS constraints.

C.3.5 Satisfy Downstream Air Traffic Flow Restrictions

These can be flow constraints that restrict either the takeoff time of specific flights (DSP – Departure Sequencing Program, EDCT – Expected Departure Clearance Time) or the takeoff time or distance between certain flights (Miles or Minutes In Trail). In all these cases, the time or distance constraints can be translated to departure sequence position constraints. This makes them similar to the fairness constraints with respect to their coupling behavior to the system design parameters.

Given the above set of arguments, the design equation in Figure C.3 now becomes:

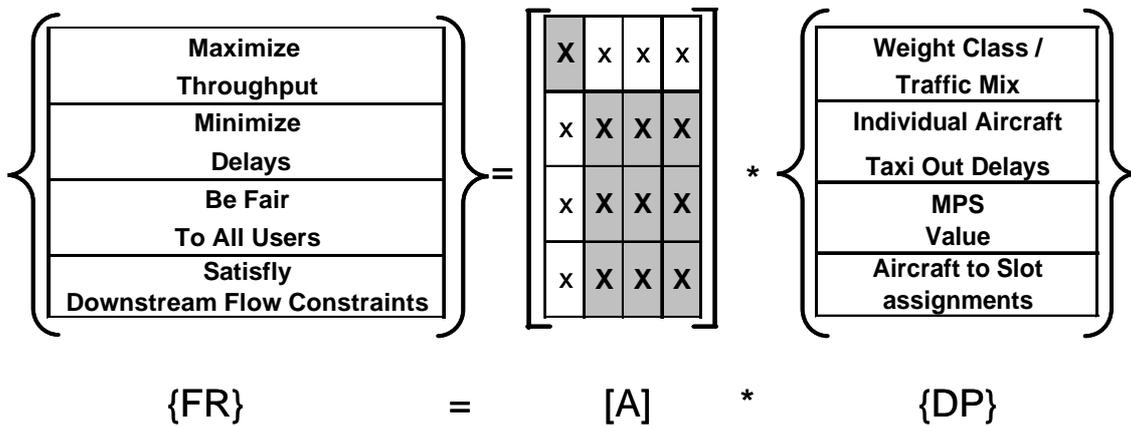


Figure C.4: ROP Design Matrix Equation – Final

Appendix D

SIMULATION OUTPUT

In the following sections a few sample results are presented in order to provide details on:

1. How Weight Class Sequences are formed and prioritized as an output of the 1st stage of the algorithm and
2. How Aircraft Class Sequences, i.e. final “aircraft to weight class slot” allocations are determined based on the minimization objective function used in the 2nd stage of the algorithm calculations.

D.1 1st Stage Output: Weight Class Sequences

In the first stage of the optimization process, departure plans are generated as sequences of weight class slots and the departure throughput performance associated with each class sequence is calculated as the inverse of the total time (in seconds) that it takes to complete all the departure

operations in the schedule. A sample subset of the results from the 1st stage of the algorithm is shown in Figure D.1 for a case with a constructed pushback schedule input of nine (9) departing aircraft. The values under each of the three rightmost columns under the heading “Time to Complete” are the times (in seconds) necessary to complete all operations of the corresponding sequence.

Schedule Index No															Time to Complete (sec)		
															Without X	With X	Stoch. With X
1	S	S	x	L	L	x	x	H	x	x	L	L	L	L	600	766.06	751.09
2	S	S	x	L	L	x	x	L	L	x	x	L	L	H	570	786.06	770.25
3	S	S	x	L	H	x	x	L	L	x	x	L	L	L	600	778.74	777.75
4	S	x	L	L	x	x	H	x	x	L	L	L	L	S	600	826.06	803.39
5	S	S	x	L	L	x	x	L	H	x	x	L	L	L	600	826.06	803.52
6	S	S	x	L	L	x	x	L	L	x	x	H	L	L	600	816.06	807.3
7	S	S	x	H	x	x	L	L	x	x	L	L	L	L	600	838.74	833.57
8	S	x	L	H	x	x	L	L	x	x	S	L	L	L	600	838.74	838.03
9	S	x	L	S	x	x	L	L	x	x	L	L	L	H	570	846.06	841.43
10	S	x	L	L	x	x	L	S	x	x	L	L	L	H	570	846.06	843.08
11	S	x	L	L	x	x	L	S	x	x	H	L	L	L	600	876.06	875.23

Figure D.1: Departure Class Schedules with a Heavy as the “extra” aircraft

The first column contains the “time to complete operations” without taking into account the time necessary for crossing operations. These values were calculated using the inter-departure separation criteria presented in Table C.2. The second column contains the “time to complete operations” after crossing aircraft have been added to each sequence. In Figure D.1, crossings are shown by the lowercase letter “x”.

The “time to complete operations” results in Figure D.1 are sorted in ascending order, i.e. from higher to lower departure runway throughput values, with respect to the third columns entries.

This column presents the results of calculating the stochastic throughput performance of each class sequence using the method presented in section 5.4.3. The results include the time to complete the last of the 9 departure operations up to the end of the last operation in the sequence when the next aircraft is cleared to take off⁵⁰. Due to wake vortex separations criteria, the end time point of the last operation depends on the weight class of the aircraft that follows. In the example results of Figure D.1, the “aircraft that follows” is fixed to be a Heavy but is not part of the aircraft window that is currently being optimized.

If the “aircraft that follows” had been fixed to be a Large or a Small, the “time to complete operations” in Figure D.1 could, in some cases, be increased and therefore, the corresponding throughput values could be decreased. For example, assuming a Large “aircraft that follows”, the “time to complete operations” in sequence 2 of Figure D.1, would be 600 sec and not 570 sec, because, based on the wake vortex separation criteria of Table C.2, the Heavy class slot at the end of sequence 2 would require a wake vortex separation delay of 120 sec behind it with a Large aircraft trailing. On the other hand, with a Heavy aircraft trailing (as presented in Figure D.1), only 90 sec of wake vortex separation are necessary.

For the remaining comments, we focus only on the results presented in the 1st and 2nd column of Figure D.1. As verified by the apparent difference in the values of these two columns, the introduction of crossings has a diminishing effect on runway throughput. Also it can be observed that the same set of crossings can change the throughput performance of certain

⁵⁰ Assuming constant pressure on the departure runway.

sequences less than others, especially when a crossing group is scheduled to cross the runway right after a heavy (H) departure. For example, we compare sequences 1 and 2 in Figure D.1. If crossings were not taken into account, one would expect sequence 2, which has the single Heavy slot at the end and needs only 570 sec to complete operations, to be preferred as a Target Class Sequence for the 2nd stage over sequence 1, which has a higher throughput (needs 600 sec to complete operations). This difference in throughput manifests because the only Heavy slot (shaded for sequences 1 and 2 in Figure D.1) is at the end of class sequence 2 and therefore introduces only 90 sec of wake vortex separation behind it (the “aircraft that follows” is a Heavy). On the other hand, the Heavy slot is situated before the end of class sequence 1 and introduces 120 sec of wake vortex separation (a Large slot follows).

Even though, in both sequences 1 and 2, the five crossings are planned in the exact same way (three groups of 1, 2 and 2 aircraft correspondingly), they have a different impact on the throughput performance of the two sequences. In the end, after crossings are included, sequence 1 proves to be a better Target Class Sequence for the 2nd stage, because it has a better throughput performance than sequence 2. This is because, in sequence 1 the third crossing group is planned (by the algorithm) to cross the departure runway right after a Heavy aircraft takes off. The 120 seconds of necessary wake vortex separation behind that Heavy departure (a Large follows) is enough time for the crossing group to be serviced, with no need for additional departure runway time to be consumed for crossings. On the other hand, in sequence 2, the third crossing group must be serviced behind a Large departure with only 60 seconds of wake vortex separation behind it. These 60 seconds are not enough for all crossings to be completed and more departure runway time is needed for the crossings in addition of the already scheduled wake vortex

separation time. Such additional runway time that is NOT used for departures reduces the throughput performance of the departure runway and makes sequence 2 less favorable than sequence 1.

The stochastic throughput results presented in the 3rd column of Figure D.1 demonstrate that accounting for possible one-position swaps of class slots in a sequence can also have an impact on throughput performance. For example, sequences 9 and 10 have the same throughput performance before and after crossings are included in the calculation (columns 1 and 2). However, the two sequences are almost identical except for one swap between two slots (the Large and Small slots that appear shaded in both sequences 9 and 10). Based on the stochastic throughput calculation method of section 5.4.3, that one swap is enough to differentiate the two sequences in terms of stochastic throughput performance and make sequence 9 more favorable than sequence 10 (column 3).

D.1.1 Introducing Crossings in the Runway Operations Plan

The total number of crossings that interact with each departure weight class sequence can vary and depends on:

- a) The exact sequencing of departure class slots, which determines the predicted “available at the runway” times for the departures and
- b) The predicted “touchdown” times for those arrivals that, based on the “available at the runway” times from (a), are expected to interact with the departures on the departure runway.

For the particular set of aircraft in the examples presented in Figure D.1, five (5) crossings were

expected to interact with the nine (9) departures in all cases, but for other departure class sequences there could be more or less crossing operations involved.

The exact positioning of crossing operations within the departure schedules was determined using the crossing logic that was presented in section 5.4.2, which depends on:

- a) The maximum allowable delay for a crossing aircraft, namely the maximum amount of time that an aircraft is left to wait at a runway cross point before it is cleared to cross, which is a test parameter whose value can vary but was set to 200 sec for the results presented here and
- b) The capacity of taxiway segments between the two parallel runways of the benchmark airport (Figure 5.1), which sets an upper limit to the number of arriving aircraft that can be accommodated between the two runways before the departure flow has to be interrupted for crossings to be serviced on the departure runway.
- c) The weight class characteristics of the arriving aircraft, which were assumed to determine the runway exit point and the taxiway that each arrival will use after landing is completed.

A set of arriving aircraft with different weight class characteristics than the ones in Figure D.1 will most likely be parsed in different crossing groups that will be accommodated among the departures in a totally different way than in this example.

The throughput of each class sequence with crossings is calculated as follows:

For each departure class slot in the sequence, a tentative “start time” can be calculated by following wake vortex separation times and crossing periods from the beginning of the sequence up to the beginning of that departure class slot. The necessary crossing periods are calculated

with the heuristic rule mentioned in section 5.6.1, i.e. 40 sec for the first aircraft to cross and 10 sec for each aircraft following. During this process, the “start times” of departure class slots are tentatively set equal to the “earliest runway availability times” of departing aircraft of the same weight class characteristics (i.e. the earliest time aircraft can join the take off queue if there is one)⁵¹. That “earliest time” depends on the gate that the earliest aircraft of that weight class is coming from and the average taxi time that it takes for that aircraft to enter the runway queue and consequently reach the runway.

If the tentative “start time” of a slot is earlier than the “earliest runway availability time” of the departing aircraft (of the same weight class) whose turn is to be used in the calculation⁵², then the “start time” for that slot has to be delayed until the corresponding departing aircraft (of the same weight class) can be available at the runway. This introduces in the schedule an additional runway “idle” time that cannot be avoided and increases the “time to complete” operations. In this case, the final throughput is reduced but the throughput performance calculation is more realistic. The case that was just described, is very likely to occur, especially if the class slot whose “start time” is being calculated, is the only one of that type in the whole sequence and the corresponding aircraft of the same weight class is also the only one of that weight class in the pool of aircraft being optimized (e.g. a class sequence with a single “Heavy” slot and an aircraft pool with only one Heavy aircraft expected to push back).

⁵¹ This was neglected in the values of the first column, hence the integer values as opposed to the real numbers in the second column.

⁵² The “earliest runway availability times” of all previous departing aircraft have already been used for the calculation of the “start time” of other departure slots earlier in the sequence.

As an example, consider the following subset of a weight class sequence:

$$S - S - H - s/s/l - S - L - L \dots$$

If the earliest time that any small aircraft from the departure group can be at the runway is 670, then, based ONLY on wake vortex separations between class slots and on landing and crossing runway occupancies, the following set of times corresponds to the start times for the class slots in the above weight class sequence:

$$670 - 730 - 790 - (\mathbf{X}) - 910 - 970 - 1030 - \dots$$

These time values can be used to calculate the time to complete all departures and therefore the departure throughput as the inverse of that time. However, if the “earliest runway availability times” for all the aircraft involved are:

Small 1: 670

Small 2: 695.5

Small 3: 891.2

Heavy 1: 820

Large 1: 1010.4

Large 2: 1050.3

then the above sequence of “slot start times” is modified as follows:

$$670 - 730 - 820 - (\mathbf{X}) - 940 - 1010.4 - 1070.4 - \dots$$

The first two start times remain unchanged because the corresponding aircraft of the same weight class have “earliest runway availability times” equal to or earlier than those times, i.e. $670 = 670$ and $695.5 < 730$. However, the next class slot needs to be occupied by a Heavy aircraft and the earliest time that the only existing Heavy aircraft can reach the runway is at 820. Therefore, the

start time of the Heavy class slot has to be delayed 30 seconds from time 790 to time 820. Similarly, the start times of all subsequent class slots are modified and the final time to complete this set of operations is $1070.4 - 670 = 400.4$ seconds, which is higher than the $1030 - 670 = 360$ seconds that were calculated based only on wake vortex separations and landing and crossing runway occupancies and therefore leads to lower departure throughput.

D.2 2nd Stage Output: Runway Operations Plans

After all generated class sequences are ordered by throughput, each of them is considered as a Target Class Sequence whose slots will be populated by specific aircraft in the 2nd stage of the algorithm to possibly yield a feasible optimal runway operations plan.

D.2.1 Output Format

As an example, consider a Target Class Sequence of fifteen (15) departing aircraft (three (3) small, ten (10) large and two (2) heavies) that interacts with sixteen (16) arriving aircraft. The only active restrictions are the maximum crossing delay and taxiway capacity constraints. Presented in Table D.1 is an example final aircraft schedule as the output of the 2nd stage of the optimization. The weight class sequence used (with lowercase letters for the crossings involved) is:

L S L s s s L L l s s L L s s s L S H s s s s L H s s s L S L

Time	Flight Number	Weight Class	Pushback Sequence Position	Arrival Schedule Position	Take off Sequence Position	Elapsed Time (min)	Crossing Delay (sec)
6:35	GAA249	L	4		1	10	
6:36	BLR665	S	1		2	15.4214	
6:37	FDX3601	L	5		3	11.4515	
6:38	FDX1464	s		13			174
	ASH5268	s		14			114
	DAL1814	s		15			54
6:39	GAA259	L	3		4	14.4865	
6:40	DAL323	L	2		5	16.5297	
6:41	CJC4830	l		16			159
	USA163	s		17			104
	KAP28	s		18			39
6:42	ACA381	L	6		6	15.7105	
6:43	BLR657	L	8		7	14.2775	
6:44	GAA417	s		19			154
	KHA716	s		20			94
	ASH5312	s		21			24
6:45	N65MJ	L	13		8	11.2665	
6:46	AA381	S	10		9	15.729	
6:47	UCA4552	H	7		10	20.2487	
6:48	CJC4770	s		22			194
	USC151	s		23			139
	UCA4503	s		24			74
	GAA284	s		25			4
6:49	DAL2503	L	11		11	18.1508	
6:50	UCA4630	H	9		12	21.0482	
6:51	USA851	s		26			129
	UCA4680	s		27			64
	USA2274	s		28			4
6:52	GAA269	L	14		13	17.3672	
6:53	USA1975	S	12		14	19.9452	
6:54	GAA468	L	15		15	19.7728	

*Total time to complete departures is 20 minutes,
 Departure Runway Throughput is 45 departures/hour
 Total Runway Throughput is 93 operations/hour
 Total number of departure position shifts is 13
 Cumulative Elapsed Time is 242.0441 minutes
 Cumulative Departure Delay is 58.7107 minutes*

Table D.1: Example Output of the Two-Stage Algorithm (Runway Operations Plan)

At 6:35, Departing flight GAA249, which is operated by a large aircraft and was number 4 in the

pushback sequence, is scheduled to take off in the first departure slot of this runway operations plan. If this happens, GAA249 will not suffer any queuing delay, since it is the first aircraft of this group scheduled to take off and its elapsed time will be 10 minutes. However, the flight scheduled to take off in the 2nd takeoff slot one minute later (at 6:36), which is flight BLR665 that is operated by a small aircraft and was the first to pushback, will experience an elapsed time of 15.4214 minutes. Part of this elapsed time will be its unimpeded taxi time (gate pushback until it joins the runway takeoff queue) and the remaining will be departure delay that BLR665 will suffer while waiting for the first aircraft (GAA249) to take off, if BLR665 arrives at the runway before GAA249 has taken off.

For all the simulation tests results from which are presented in Chapter 6, it was assumed that:

- The maximum allowable crossing delay per aircraft is 200 seconds;
- The taxiway space between the two parallel runways of this configuration can accommodate either four small, or two large or one heavy and one small aircraft; and
- All small arrivals exit the runway early and occupy only the first cross-point, while all the large and heavy arrivals roll on to the second runway cross-point. In real world operations this assumption can be relaxed. For example, in case the arrivals schedule mainly consists of small aircraft and there is departure pressure on the runway, small landing aircraft can also be sent to the second cross-point provided that the maximum crossing delay constraint is not violated for any arrival. This way, clearing a group of arrivals to cross the departure runway can be deferred for a while longer, in order to avoid performing crossings just because one of the two cross points is saturated, while the other one is mostly empty.

Under these assumptions, the crossing parts in the above example output can be read as follows: At 6:38, three small aircraft are scheduled to cross the runway. Arriving flights FDX1464, ASH5268 and DAL1814, become crossing requests on the departure runway. Based on their expected touchdown times and depending on the time the departing aircraft were expected to reach the runway, these three crossings were scheduled by the crossing logic to cross the runway right after the 3rd departure (FDX3601). Their corresponding touchdown positions within this arrival group and the crossing delay that each of them has to absorb are also listed. Even though there was another aircraft at the runway available for takeoff behind the 3rd departure and even though there was available taxiway space for one more small aircraft (up to a total of four small), the crossing group of the three small aircraft that had already accumulated between the two runways had to be serviced on the departure runway before more departures were allowed. That is because, if priority had been given to another departure before the crossings were released the maximum crossing delay constraint of 200 seconds would have to be violated for the first of the three aircraft waiting to cross.

D.2.2 Allocation of Aircraft to Weight Class Slots

Every time the planning simulation is run, at the end of the 1st stage a set of weight class sequences with crossings is created and its members are ordered in ascending order with respect to the departure throughput that is achieved with each of them. In most cases, there are subgroups of weight class sequences that are characterized by the same departure throughput even though the sequence of weight class slots is unique in each of them. For example, the following two class sequences with crossings:

L L L s s s L L s s l L L s s s L L H s s s s L H s s s S S S

and

L S L s s s L L s s l L L s s s L S H s s s s L H s s s L S L

may yield exactly the same departure throughput⁵³.

In the 2nd stage of the simulation, a search for an optimal feasible “aircraft to class slot allocation” is performed within each subgroup of weight class sequences with the same throughput. If an optimal feasible solution is reached based on a particular weight class sequence within a subgroup, it is not necessarily the best solution overall. It is of course the optimal “aircraft to class slot allocation” achieved if the particular weight class sequence is followed but there are, as we said, other weight class sequences that can yield the same departure throughput but with a different weight class sequence and therefore a different “aircraft to slot allocation”. So, as soon as an optimal feasible solution is found within a weight class sequence subgroup, the algorithm tests every other weight class sequence within that subgroup for a solution. Some class sequences may provide a feasible solution and some may not.

The question that arises from the above is: “*what should the objective function of the 2nd stage be in order to successfully differentiate between optimal feasible solutions that yield the same departure throughput?*” Four different objective function options are presented here:

a) A linear combination (equation D.1) of minimizing the cumulative elapsed time⁵⁴ for all

⁵³ In fact, in the example of which the solution was presented in section D.2.1, they actually do.

⁵⁴ As it was defined earlier in section 6.3.1.

departing aircraft involved in the planning and at the same time maximizing “fairness” among aircraft by minimizing the cumulative number of position shifts between pushback and takeoff that the departures have to undergo, i.e.

min (Cumulative Elapsed Time + Cumulative Number of Position Shifts)

$$\min \left(a_1 * \sum_{i=1}^{N_D} (Toff_i(p_{TOi}) - Tpb_i)^{k_D} + a_2 * \sum_{i \in N_F} (p_{TOi} - p_{PBi}) \right) \quad (D.1)$$

where for each aircraft i , p_{TOi} and p_{PBi} are its takeoff and pushback position correspondingly, $Toff_i$ is the actual takeoff time and Tpb_i the scheduled gate pushback time and N_F is the set of departing aircraft (subset of N_D) which, after the slot allocations, they are penalized by being moved backwards in the takeoff sequence (i.e. they have positive position shift).

- b) A linear combination (equation D.2) of minimizing the cumulative departure delay for all departing aircraft involved in the planning and the cumulative number of position shifts between pushback and takeoff that the departures have to undergo:

min (Cumulative Departure Delay + Cumulative Number of Position Shifts)

$$\min \left(a_1 * \sum_{i=1}^{N_D} (Toff_i(p_{TOi}) - Eoff_i)^{k_D} + a_2 * \sum_{i \in N_F} (p_{TOi} - p_{PBi}) \right) \quad (D.2)$$

The weight parameters a_1 and a_2 can be used to vary the impact that each of the two terms of the objective function in equations (D.1) and (D.2) will have on the final solution. In the scope of this work, when both terms were used in the objective function, both a_1 and a_2 were assumed to be equal to 1. If a_1 is set to 1 and a_2 is set to 0, then two more time-based objective functions are generated, which only optimize cumulative elapsed times or departure delays:

c) min (Cumulative Elapsed Time), i.e.

$$\min \left(\sum_{i=1}^{N_D} (T_{off_i}(p_{TOi}) - Tpb_i)^{k_D} \right) \quad (D.3) \text{ and}$$

d) min (Cumulative Departure Delay), i.e.

$$\min \left(\sum_{i=1}^{N_D} (T_{off_i}(p_{TOi}) - E_{off_i})^{k_D} \right) \quad (D.4)$$

Each of these objective functions was applied in the example schedule of fifteen (15) departing aircraft for which results were shown in Table D.1 and the collective performance results are presented in Table D.2.

This example was chosen in order to demonstrate, as in Table D.2, how the choice of objective function for the 2nd stage can affect various characteristics of the final optimal runway operations plan, even if that final plan has exactly the same throughput and delay performance in all cases. It seems that, among the four solutions presented in Table D.2, the ones resulting from objective functions (D.1) and (D.2) should be the most desirable for airline users. They both plan runway operations with a maximum throughput and minimum cumulative elapsed times and cumulative departure delays correspondingly, but at the same time they both introduce to the aircraft group the least deviation from a First Come First Serve (FCFS) “pushback to takeoff” policy, i.e. a total number of 13 delaying position shifts. That is because they were both generated, under a strict “fairness among all users” scenario, i.e. with a “fairness” term included in the objective function. In addition, air traffic controllers are more likely to prefer these solutions, because they are the least workload-intensive in terms of aircraft sequencing.

Pushback Position	Take off Pos. (Obj. Func. D.1)	Take off Pos. (Obj. Func. D.2)	Take off Pos. (Obj. Func. D.3)	Take off Pos. (Obj. Func. D.4)
1	2	2	14	9
2	5	5	5	4
3	4	4	4	5
4	1	1	1	1
5	3	3	2	2
6	6	6	3	3
7	10	12	10	12
8	7	8	7	7
9	12	10	12	10
10	9	9	15	8
11	11	7	6	6
12	14	14	13	15
13	8	11	8	11
14	13	13	9	14
15	15	15	11	13
Dep. Runway Throughput	45 departures / hr			
Total Runway Throughput	93 ops / hr (16 crossings)			
Cumulative Dep. Position Shifts	13	13	29	21
Cumulative Dep. Delay	58.7107 min	58.7107 min	58.7107 min	58.7107 min
Cumulative Elapsed Time	242.0441 min	242.0441 min	242.0441 min	242.0441 min

Table D.2: Example - Ground Traffic Performance Metrics and “Aircraft to Slot” Assignments under the same pushback schedule input and with respect to different 2nd stage objective functions (no air traffic flow restrictions active)

The runway operations plan outlined in Table D.1 is the one in the 2nd column of Table D.2 and was created using objective function (D.1). Note that, both that plan and the one in the 3rd column of Table D.2 were generated based on the same weight class sequence with crossings:

L S L s s s L L s s l L L s s s L S H s s s s L H s s s L S L

This means that, for this particular problem, whether cumulative elapsed times (objective function. D.1), or cumulative departure delays (objective function. D.2) are minimized, as long as the “fairness” term is part of the objective function, the feasible optimal solution generated by the Branch & Bound solver is based on the same weight class sequence. This class sequence

however, is used for two slightly different “aircraft to slot allocations” in the final solution because a different time-based performance metric is optimized in each case.

The “fairness” term that was introduced in the objective functions (D.1) and (D.2) divides the focus of the objective function between the time-based “cumulative departure delay” or “cumulative elapsed time” minimization and the workload / fairness-based “cumulative number of aircraft position shifts” minimization. That way the Branch & Bound solver in the 2nd stage of the algorithm is “forced” towards a more desirable direction among all optimal solutions with the same throughput and delay / elapsed time performance. This can be seen in the results of the 4th and 5th column of Table D.2, in which, objective functions (D.3) and (D.4) were used and minimization of the “cumulative number of aircraft position shifts” was not performed. The throughput and delay / elapsed time performance is identical to the one of the runway operations plans in the first two columns of Table D.2, but the cumulative number of delaying position shifts is much higher in both cases. That is because, since the “position shifts” minimizing term is not part of the objective function, the Branch & Bound solver, on one hand can identify the optimal feasible solution in all weight class sequences tested, but on the other hand, it does not have a way to differentiate between all the optimal feasible solutions from all the tested class sequences and choose one that has additional better characteristics. Note that, those two runway operations plans presented in the last two columns of Table D.2 were generated based on the following two different weight class sequences with crossings:

L L L s s s L L s s l L L s s s L L H s s s s L H s s s S S S

for objective function (D.3) and

LLSssLLSs1LLSssSSHssssLHssSLLS

for objective function (D.4).

Appendix E

CONVERGENCE OF THE MONTE CARLO SIMULATION

The conclusions of the stability analysis in 6.5 were based on running the Monte Carlo simulation in the scenario WITH ADAPTIVE Planning (Figure 6.22) only 500 times for each set of uncertainty parameters that was tested. Limiting the number of Monte Carlo runs was necessary in the interest of simulation runtime, because each run is basically the generation of a new runway operations plan which takes a few minutes to be completed and therefore, a series of consecutive runs for each simulation scenario takes a very long time.

However, the 500-run limit value was an educated choice. Convergence analysis of the Monte Carlo simulation proved that the statistical results do not change significantly if more than 500 runs were performed. The convergence analysis consisted of running a few simulation scenarios for up to 3,000 times and checking how the stability results, as well as the throughput and delay performance of the generated runway plans evolved as more runs were being completed. The results for one of these simulation scenarios (4 aircraft do not push back on time, pushback delay

Appendix E: Convergence of the Monte Carlo Simulation

standard deviation = 120 sec) are presented in Figure E.1, Figure E.2 and Figure E.3. This scenario is representative of a high level of uncertainty, because, each time, there are 4 aircraft that have their scheduled pushback times affected by uncertainty and also the standard deviation value of 120 sec is much higher than the time interval of 60 sec between consecutive pushbacks in the “nominal” schedule.

It is evident that the values of the two stability metrics that were used in the stability analysis and the throughput and delay characteristics of the generated plans do not vary significantly for any number of Monte Carlo runs above 500. This led us to believe that, limiting the number of Monte Carlo runs to 500 for each simulation scenario did not bias the conclusions of the stability and robustness analysis.

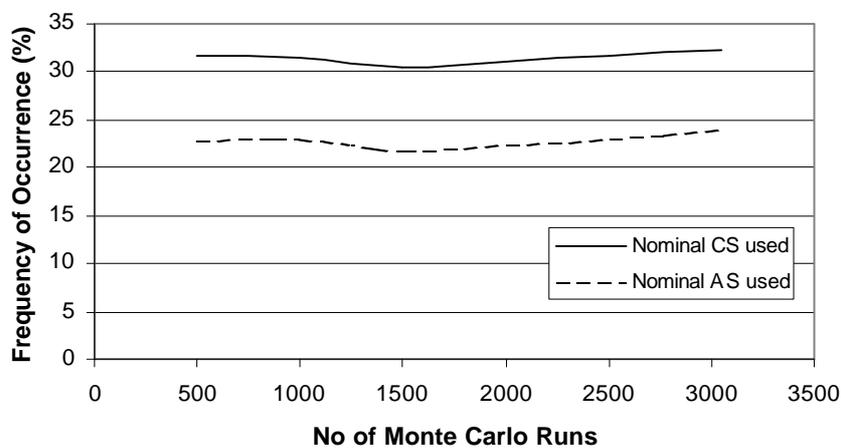


Figure E.1: Simulation Convergence Analysis – Stability Metrics

Appendix E: Convergence of the Monte Carlo Simulation

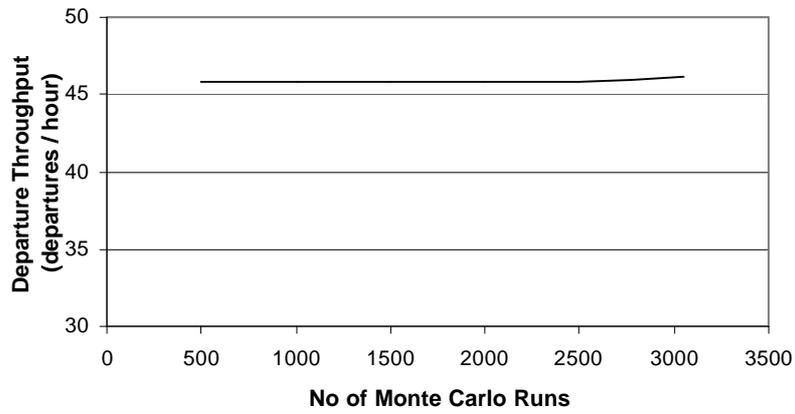


Figure E.2: Simulation Convergence Analysis – Departure Throughput

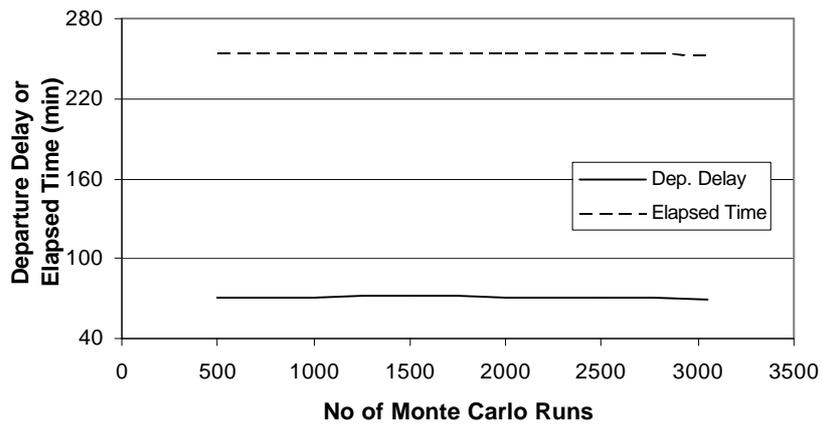


Figure E.3: Simulation Convergence Analysis – Departure Delay & Elapsed Time

Appendix F

STANDARD ERROR ANALYSIS

F.1 Effect of Air Traffic Flow Restrictions

	WITHOUT Flow Restrictions		WITH Flow Restrictions	
	WITHOUT Planning	WITH Planning	WITHOUT Planning	WITH Planning
Dep. Runway Throughput (departures / hour)	0.06	0.048	0.02	0.019
Total Runway Throughput (operations / hour)	0.06	0.045	0.02	0.02
Cumulative Dep. Delay (min)	0.23	0.25	0.11	0.1
Cumulative Elapsed Time (min)	0.3	0.3	0.13	0.12

Table F.1: Standard Error of the Mean Values of Performance Metrics for scenarios with and without Planning under Restricted and Unrestricted Traffic Conditions (re: Figure 6.11, Figure 6.12, Figure 6.13, and Figure 6.14)

<i>Affected By ...</i>	WITHOUT Planning			WITH Planning		
	<i>No MIT</i>	<i>One MIT</i>	<i>Two MIT</i>	<i>No MIT</i>	<i>One MIT</i>	<i>Two MIT</i>
Dep. Runway Throughput (departures / hour)	0.03	0.02	0.03	0.016	0.02	0.06
Total Runway Throughput (operations / hour)	0.03	0.02	0.03	0.018	0.02	0.06
Cumulative Dep. Delay (min)	0.14	0.14	0.28	0.1	0.13	0.4
Cumulative Elapsed Time (min)	0.16	0.15	0.3	0.11	0.14	0.4

Table F.2: Standard Error of the Mean Values of Performance Metrics for a Restricted Scenario with and without Planning (re: Figure 6.15, Figure 6.16, Figure 6.17 and Figure 6.18)

F.2 Effect of Pushback Timing

<i>Schedule Type</i>	WITHOUT Planning			WITH Planning		
	<i>Original</i>	<i>Bunch Front All</i>	<i>Bunch Front Partial</i>	<i>Original</i>	<i>Bunch Front All</i>	<i>Bunch Front Partial</i>
Dep. Runway Throughput (departures / hour)	0.06	0.023	0.024	0.048	0.044	0.046
Cumulative Dep. Delay (min)	0.23	0.13	0.13	0.25	0.26	0.26
Cumulative Elapsed Time (min)	0.3	0.16	0.16	0.3	0.31	0.31

Table F.3: Standard Error of the Mean Values of Performance Metrics for Schedules with Varying Pushback Time Distributions (re: Figure 6.31, Figure 6.32 and Figure 6.33)

<i>Schedule Type</i>	WITHOUT Planning			WITH Planning		
	<i>Regular</i>	<i>Bunch Back Partial</i>	<i>Bunch Back All</i>	<i>Regular</i>	<i>Bunch Back Partial</i>	<i>Bunch Back All</i>
Dep. Runway Throughput (departures / hour)	0.041	0.06	0.025	0.036	0.049	0.028
Cumulative Dep. Delay (min)	0.2	0.27	0.25	0.21	0.26	0.26
Cumulative Elapsed Time (min)	0.23	0.32	0.28	0.25	0.31	0.32

Table F.4: Standard Error of the Mean Values of Performance Metrics for Schedules with Varying Pushback Time Distributions (re: Figure 6.31, Figure 6.32 and Figure 6.33)

F.3 Effect of Demand Weight Class Mix

	WITHOUT Planning			WITH Planning		
<i>Weight Class Mix (% Heavies, % Large, % Small)</i>	<i>1-14-0</i>	<i>2-10-3</i>	<i>5-5-5</i>	<i>1-14-0</i>	<i>2-10-3</i>	<i>5-5-5</i>
Dep. Runway Throughput (departures / hour)	0.038	0.041	0.047	0.039	0.036	0.04
Cumulative Dep. Delay (min)	0.19	0.2	0.26	0.19	0.21	0.28
Cumulative Elapsed Time (min)	0.24	0.23	0.31	0.24	0.25	0.33

Table F.5: Standard Error of the Mean Values of Performance Metrics for Schedules with Evenly-Spaced Pushback Times (Regular) and Varying Demand Weight Class Mix (re: Figure 6.34, Figure 6.35 and Figure 6.36)

Appendix G

RUNWAY OPERATIONS PLANNING IN ONE STAGE

The Runway Operations Planning algorithm proposed in Chapter 3 may be designed in two stages as it was derived based on the decomposition methodology also introduced in Chapter 3, but it should be noted that a two-stage approach is not a unique way to address the problem of Runway Operations Planning. In fact, an earlier attempt to address a very similar problem with a different solution methodology was presented in [2]⁵⁵. The problem was separated in two minimization subtasks in order to become simpler, but the solution was still developed in a “*one-stage*” *optimization routine* that considered all objectives and constraints at the same time. That single stage method contains the following two main subtasks:

- Assuming a time-based objective function $Q(\mathbf{t})$, the optimal takeoff time (schedule) vector

⁵⁵ A joint research publication between MIT and DLR, but credit for the mathematical formulation of this one-stage approach belongs to DLR.

\mathbf{t}^* for a specific aircraft sequence s can be calculated as:

$$\mathbf{t}^* = \mathbf{t}^*(s) = \arg \min_{\mathbf{t} \in T} Q(\mathbf{t}) \quad (\text{G.1}) \text{ and}$$

- Knowing the optimal takeoff time vector \mathbf{t}^* for each sequence, the optimal sequence s^* results from minimizing the values $Q(\mathbf{t}^*)$ among all member sequences of the set S of feasible sequences:

$$s^* = \arg \min_{s \in S} Q(\mathbf{t}^*(s)) \quad (\text{G.2})$$

Therefore, the following equation:

$$\mathbf{t}^{**} = \mathbf{t}^*(s^*) = \arg \min_{s \in S} \min_{\mathbf{t} \in T} Q(\mathbf{t}) \quad (\text{G.3})$$

yields the global optimum vector of takeoff times. The first minimization subtask can be performed by sequentially determining the takeoff times as the earliest feasible release times subject to a set of constraints, while, for the selection of the best sequence in the second subtask, a search tree can be used.