Tractability Through Approximation: A Study of Two Discrete Optimization Problems

by

Amr Farahat

B.Sc., The American University in Cairo (1994)
M.Sc., Lancaster University (1996)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2004

Signature of Author ....
Sloan School of Management
July 12, 2004

Certified by ....
Cynthia Barnhart
Professor, Civil and Environmental Engineering and Engineering Systems
Thesis Supervisor

Accepted by ....
James B. Orlin
Edward Pennell Brooks Professor of Operations Research
Co-Director, Operations Research Center

Tractability Through Approximation:

A Study of Two Discrete Optimization Problems

by

Amr Farahat

Submitted to the Sloan School of Management
on July 12, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

This dissertation consists of two parts. In the first part, we address a class of weakly-coupled multi-commodity network design problems characterized by restrictions on path flows and 'soft' demand requirements. In the second part, we address the abstract problem of maximizing non-decreasing submodular functions over independence systems, which arises in a variety of applications such as combinatorial auctions and facility location. Our objective is to develop approximate solution procedures suitable for large-scale instances that provide a continuum of trade-offs between accuracy and tractability.

In Part I, we review the application of Dantzig-Wolfe decomposition to mixed-integer programs. We then define a class of multi-commodity network design problems that are weakly-coupled in the flow variables. We show that this problem is NP-complete, and proceed to develop an approximation / reformulation solution approach based on Dantzig-Wolfe decomposition. We apply the ideas developed to the specific problem of airline fleet assignment with the goal of creating models that incorporate more realistic revenue functions. This yields a new formulation of the problem with a provably stronger linear programming relaxation, and we provide some empirical evidence that it performs better than other models proposed in the literature.

In Part II, we investigate the performance of a family of greedy-type algorithms to the problem of maximizing submodular functions over independence systems. Building on pioneering work by Conforti, Cornuéjols, Fisher, Jenkyns, Nemhauser, Wolsey and others, we analyze a greedy algorithm that incrementally augments the current solution by adding subsets of arbitrary variable cardinality. This generalizes the standard best-in greedy algorithm, at one extreme, and complete enumeration, at the other extreme. We derive worst-case approximation guarantees on the solution produced by such an algorithm for matroids. We then define a continuous relaxation of the original problem and show that some of the derived bounds apply with respect to the relaxed problem. We also report on a new bound for independence systems. These bounds extend, and in some cases strengthen, previously known results for standard best-in greedy.

Thesis Supervisor: Cynthia Barnhart
Title: Professor, Civil and Environmental Engineering and Engineering Systems

Acknowledgments

I am deeply indebted to my advisor, Professor Cynthia Barnhart, for unwavering support, guidance, and encouragement. Cindy's mentorship extended beyond the duties of research to provide genuine support for the intellectual growth and academic success of her students. Her emphasis on balance of theory and practice, of research and mentorship, and of work and family, provides a role model of an academic that I can only strive to approximate in my own career.

I am grateful to Professors Tom Magnanti and Andreas Schulz for providing generous feedback as thesis committee members, and for the active roles they played at the Operations Research Center. Professor Schulz was instrumental in developing my enthusiasm for combinatorial optimization, and Part II of this thesis was initially motivated by discussions with him.

Professor Georgia Perakis provided me with valuable teaching opportunities, and was a constant source of encouragement all along. I would also like to thank Professors Arnie Barnett and Jim Orlin for all their support and advice. I am proud to have been part of the OR community at MIT, largely because of the strength and humanity of its faculty.

Special thanks to Maria Marangiello, Veronica Mignot, Paulette Mosley, and Laura Rose for their superb administrative assistance, and for their friendship.

I am grateful to Steve Kolitz and Rina Schneur for providing constant reminders of the practical value of optimization. I also wish to acknowledge the financial support of the Draper Laboratory and of UPS.

I am very fortunate to have an exceptional group of friends who made this journey enjoyable and stimulating. In particular, my gratitude goes to Biova Agbokou, Margrét Bjarnadóttir, Sanne de Boer, Amy Cohn, Nagi Elabbasi, Pavithra Harsha, Mahmoud Hussein, Soulaymane Kachani, Laura Kang, Susan Martonosi, Heather Mildenhall, Mohamed Mostagir, Neema Sofaer, Nico Stier Moses, Ping Xu, and Hesham Younis.

My stay at MIT would not have been the same without my brother, Waleed. I am very fortunate to have shared the past few years with him as students here in Cambridge. I thank him for always being there, even when I was way too busy focused on my own work. My sister, Zeinab, though not with me in Cambridge, was always with me in spirit. I am proud to be your brother.

Finally, I am eternally grateful to my parents for the values they taught me. I dedicate this dissertation to them for having dedicated their lives to their children.

Contents

I Decomposition of Weakly-Coupled Network Design Problems **13**

1 Introduction to Part I **15**

2 Dantzig-Wolfe Decomposition for Mixed-Integer Programs **19**

2.1 Extended Reformulations 19

2.2 Literature Review 20

2.3 The Master Formulation 22

2.4 Analysis 25

2.4.1 Relation Between INITIAL and CONV 26

2.4.2 Relation Between CONV and MASTER 27

2.4.3 Decomposable Systems 30

2.5 Solution Algorithms 31

2.5.1 Solving the Linear Programming Relaxation of the Master Formulation 31

2.5.2 Solving the Master Formulation to Integrality 32

3 A Class of Weakly-Coupled Network Design Problems **33**

3.1 Problem Statement 33

3.2 Computational Complexity 36

3.3 WNDP with Coupling Capacity Constraints 41

3.4 Approximation Step 42

3.4.1 Partitions 42

3.4.2 Revenue Allocation 43

3.4.3 Decomposition 44

3.4.4 Analysis 45

3.5 Reformulation Step 54

3.6 Solution Procedure 56

4 Application: Airline Fleet Assignment with Enhanced Revenue Modeling 59

4.1 Introduction 59

4.1.1 The Fleet Assignment Problem 59

4.1.2 Literature Review 61

4.1.3 Motivation 62

4.1.4 Outline 63

4.2 Initial Formulation 63

4.2.1 Generic Model 63

4.2.2 FAM 65

4.2.3 IFAM 67

4.3 Extended Reformulation 68

4.3.1 Flight Leg / Itinerary Classification Schemes 69

4.3.2 Projections 69

4.3.3 Approximation Step 69

4.3.4 Reformulation Step 74

4.4 Proof-of-Concept Results 77

5 Part I Summary and Future Directions 81

5.1 Summary of Contributions 81

5.2 Future Directions 81

Part I Bibliography 82

II Performance of Greedy Algorithms for Maximizing Submodular Functions over Independence Systems 89

6 Introduction to Part II 91

6.1 Problem Statement 93

6.2 Preliminaries 96

6.3 Applications 98

6.3.1 Combinatorial Auctions 99

6.3.2 Bank Account Location 100

6.3.3 Channel Assignment in Cellular Networks 101

6.4 Standard Greedy Algorithm 102

7 Literature Review 105

7.1 Complexity Results 106

7.2 Performance Bounds 106

7.2.1 Linear Functions 107

7.2.2 Nondecreasing Submodular Functions 108

7.2.3 Variants of STANDARD GREEDY 109

7.3 Hardness of Approximation 110

8 Generalized Greedy Algorithm 111

9 Analysis of Bounds for Matroids 115

9.1 Comparison of Bounds 116

9.2 Matroid Bound 117

9.3 Selection of L_i 122

9.4 Application to STANDARD GREEDY 123

10 Extensions 125

10.1 Continuous Extension 125

10.2 Independence Systems 134

11 Part II Summary and Future Directions **137**

11.1 Summary of Contributions 137

11.2 Future Directions 138

Part II Bibliography **138**

List of Figures

3-1 Transformation used in proving NP-completeness of $WNDP'$. 40

3-2 Example showing duality gap for Problem DUAL(Π). 48

3-3 Proposed iterative solution algorithm for solving WNDP+. 57

3-4 Progress path of solution algorithm 58

4-1 An example of a time-line network involving two airports. 65

7-1 A hierarchy of the different classes of feasible sets. $\mathcal{F}1 \longrightarrow \mathcal{F}2$ indicates that $\mathcal{F}1 \subset \mathcal{F}2$. 106

List of Tables

4.1 Representative savings due to the parsimonious approach. 77
4.2 SFAM test instance: summary statistics. 77
4.3 A comparison between FAM, IFAM and SFAM results. 78

7.1 Complexity results for Problem (P) 106
7.2 Approximation guarantees for Standard Greedy when applied to Problem (P) 107

Part I

Decomposition of Weakly-Coupled Network Design Problems

Chapter 1

Introduction to Part I

We address the optimal design of capacitated networks serving multiple commodities simultaneously. Commodities can be physical goods, utility supplies, or information flows. The objective is to determine how much capacity to install on each link of the network and how to route commodity flows, subject to various demand and operational constraints, in order to maximize some well-defined objective function such as net revenue. A formal definition of this problem is provided in Chapter 3. The wide range of applications that can be cast as multi-commodity network design problems makes this class of problems one of the most practically significant in the field of optimization. It also embodies many of the core theoretical and computational issues underlying more general optimization problems. For good expositions of different types of network design problems, their applications, and various methods used to solve them, the reader is referred to the surveys by Magnanti and Wong [MaW84] and Minoux [Min89] and to the handbooks edited by Ball et al. [BMM95a, BMM95b].

Basic versions of the network design problem are difficult to solve in a theoretical as well as in a practical sense. Theoretically, many of the simplest versions are NP-hard. Practically, network design problems give rise to mixed-integer programming formulations that have notoriously weak linear programming (LP) relaxations, and are, therefore, elusive to solve using standard LP-based branch-and-bound methods.

A major cause of LP fractionality arises from the interplay between the discrete (capacity design) variables and the continuous (flow) variables. Because capacity provision is penalized in the objective function, LP relaxations of standard network design formulations provide 'just enough' capacity on each arc to accommodate flow. The simple fact that it is unlikely for flow value on any given arc to coincide with any one of the discrete capacity levels available for that arc implies that fractionality permeates the LP solution, rendering branch-and-bound schemes ineffective. Not surprisingly, much of the research on network design problems has the goal of devising generic methods to strengthen LP relaxations. This research draws from and contributes to ideas in the more general setting of large-scale mixed-integer programming. The classical approach of Dantzig-Wolfe (DW) decomposition, developed originally for linear programs, is now emerging as an attractive approach to strengthening the relaxation of integer programs. We view DW decomposition for integer programs as essentially a formalism for the enumeration of partial solutions in discrete sets.

Our objectives in Part I of this dissertation are three-fold, corresponding to three levels of generality:

a) In Chapter 2, we provide an exposition of selected literature pertaining to the generation, comparison, and solution of alternative reformulations of mixed-integer programs. We focus on DW type approaches, and provide a basic exposition that forms a foundation for the following chapters.

b) In Chapter 3, we specialize our discussion to network design problems, and argue that DW decomposition methods lead to strong formulations by eliminating the flow variables. This comes at the expense of increasing the number of integer variables. Hence, for this method to work in practice, we need to consider problem instances characterized by a weak coupling of the flow variables. This is essentially equivalent to problems with restrictions on path flows. We analyze the complexity of a core class of weakly-coupled problems and present algorithms that specialize the techniques of DW reformulation to deal with such instances.

c) Finally, in Chapter 4, we apply the ideas developed in Chapter 3 to the specific problem of airline fleet assignment with the goal of creating models that incorporate more realistic revenue functions. We show how the DW reformulation approach yields a new formulation of the problem that is more general and performs better than those previously proposed in the literature. Limited computational results based on data drawn from a major airline show significant monetary savings.

Presenting the DW reformulation approach through a continuum of problems ranging from the most general to the very specific helps 'demystify' some of the ad hoc mechanics of specific applications and enables the identification of instances that can benefit from the same approach. It also reflects our opinion that for an implementation of a general method to succeed in practice it often has to be tailored to the specifics (structure and data) of a particular application. We also favor solution methods that present a family of trade-offs between accuracy (degree of approximation) and tractability (computational burden).

Chapter 2

Dantzig-Wolfe Decomposition for Mixed-Integer Programs

In this chapter we first position DW decomposition within the larger picture of 'extended reformulations'. We then outline the literature pertaining to DW methods for linear and pure integer programs. The *master formulation* is developed in Section 2.3 followed, in Section 2.4, by an analysis of its LP strength. Finally, in Section 2.5, we conclude by outlining solution procedures for solving the master formulation.

Most of the material in this chapter is a synthesis of standard treatments of DW decomposition. We provide a summary treatment here, including essential proofs, not for novelty, but for completeness and as foundation for the rest of Part I. The reader is referred to cited references for more detailed expositions.

2.1 Extended Reformulations

It is well known that a given problem can have different formulations that are all logically equivalent yet differ significantly from a computational point of view. This has motivated the study of systematic procedures for generating, solving, and comparing alternative formulations.

Broad classifications of these procedures are outlined by Geoffrion [Geo70a, Geo70b], and Martin [Mar99]. The framework outlined by Martin distinguishes between methods that operate in the original space of variables and those that do not. The former are essentially cutting plane methods while the latter are variable redefinition methods. Cutting plane methods are aimed at finding better polyhedral approximations of the convex hull of feasible solutions. Variable redefinition methods have been subdivided by Martin [Mar99] into:

1. *Projection methods.* These are methods that remove variables from the original formulation, usually at the expense of adding an exponential number of new constraints. Examples include Fourier-Motzkin elimination and Benders' decomposition.

2. *Inverse projection methods.* These are methods that remove constraints from the original formulation, usually at the expense of adding an exponential number of variables.

Inverse projection methods include DW reformulation which is the focus of this review. From an intuitive viewpoint, the new variables can be described as *composite variables*; that is, variables expressing multiple decisions or the enumeration of partial solutions. From a polyhedral viewpoint, the added variables are convexity weights associated with finite generators of the polyhedral or integer set defined by the removed constraints. These formulations are closely tied to solution algorithms that exploit their special structure; the most popular being delayed column generation and, its extension to integer problems, branch-and-price.

2.2 Literature Review

We outline the literature on the theory and practice of DW methods for linear and integer programs. For an exposition of this and other large-scale optimization techniques the reader is referred to Martin [Mar99].

First we review DW decomposition for linear programs. The pioneering work in this area is that of Ford and Fulkerson [FoF58], Dantzig and Wolfe [DaW60], and Gilmore and Gomory [GiG61, GiG63]. See Lasdon [Las70] for a classic survey. A number of variations and extensions to the basic DW scheme have been proposed. One notable extension is by Todd [Tod90] who established an interior point framework. Aside from storage space advantages of DW implementations, most computational experience reported in the literature for linear programs has failed to find conclusive evidence of running time merits. Ho [Ho87] attributes this to the lack of sophistication of reported implementations of DW methods in comparison to the highly sophisticated implementations of commercial LP codes. He reports experiments where advanced implementations of DW yield promising results. The method remains theoretically appealing (because of its economic interpretations and as a generalization of the Simplex method), but has not enjoyed widespread applicability as a solution technique for large-scale linear programs. However, its closely related technique of column generation has enjoyed much success as an independent solution method for problems where the initial (or natural) formulation exhibits a huge number of variables.

The ideas underlying DW reformulation can be readily extended to pure integer programs; see Vanderbeck and Wolsey [VaW96], Barnhart et al. [BJN98], and Vanderbeck [Van00]. In contrast to linear programs, this reformulation scheme is particularly attractive for integer programs because of the tightening of the LP relaxation it generally achieves. If the resulting master formulation has very little fractionality at the root node then the problem can be solved as a linear program using column generation methods followed by some heuristic rounding-off. In cases where fractionality is excessive, incorporating column generation into the branch-and-bound tree is possible though not straightforward. Combining column generation and branch-and-bound is called branch-and-price and is the subject of recent and current research. An example is given by Vanderbeck and Wolsey [VaW96]. We briefly discuss branch-and-price in Section 2.5. For a more thorough exposition of branch-and-price the reader is referred to Barnhart et al. [BJN98].

It should be pointed out that DW reformulation approaches are equivalent (perhaps in a dual sense) to Lagrangian relaxation methods. For details of the latter and proofs of equivalence, the reader is referred to Magnanti, Shapiro, and Wagner [MSW76], Martin [Mar99], and Wolsey [Wol98].

We briefly note that an extended reformulation strategy has been suggested by Sherali and Adams [ShA90, ShA94] for binary mixed integer programs having nonlinear objective functions (as long as these functions are polynomial in the integer variables). It yields a hierarchy of relaxations spanning the range from the standard LP relaxation to the convex hull of the feasible region.

Applications of DW reformulation and column generation are pervasive in the literature. Transportation and logistical applications have, in particular, been fruitful areas for the application of these techniques. Desrosiers et al. [DDS95] present an extensive exposition of these techniques in the context of routing and fleet scheduling problems with complex spatial and temporal constraints (fleets being aircraft, trucks, buses, railway locomotives etc.). Specific applications include Appelgren [App69, App71], Desrosiers et al. [DSD84], Desrochers et al. [DDS92], and Dumas et al. [DDS91]. Airline applications include Crainic and Rousseau [CrR87], Desrochers and Soumis [DeS89], Lavoie et al. [LMO89], Desaulniers et al. [DDD97], and Gamanche et al. [GSM99]. Armacost et al. [ABW02] and Barnhart et al. [BKK02] present service network design applications. Applications to multi-commodity network flows include Jones et al. [JLF93] and Barnhart et al. [BHJ95, BHV00]. In all these applications, DW methods offer an approach to deal with large problem instances or to incorporate more realistic assumptions in models of complex problems. Degree of success depends on the level of fractionality of the master LP and the effectiveness in solving the pricing subproblems. In some applications, we have observed that degeneracy can be particularly problematic in solving master LPs.

2.3 The Master Formulation

Consider the following mixed-binary optimization problem which we refer to as the *initial formulation (*INITIAL*)*:

$$\max_{x,y} \; z(x,y) := f(x) + l(y) \tag{2.1}$$

$$\text{subject to}$$

$$Ax + Gy \leq b, \tag{2.2}$$

$$(x,y) \in S := P \cap (\mathcal{B}^n \times \mathcal{R}^p). \tag{2.3}$$

where:

- $x \in \mathcal{B}^n$ is a vector of binary decision variables;

- $y \in \mathcal{R}^p$ is a vector of continuous decision variables;
- $f : [0,1]^n \to \mathcal{R}$ is a concave function (concavity is for part (iv) of Proposition 2.2 to hold. Otherwise, any finite nonlinear function can be assumed);
- $l : \mathcal{R}^p \to \mathcal{R}$ is an affine function;
- P is a bounded polyhedron;
- $Ax+Gy \leq b$ are m linear *coupling* constraints. A, G, and b are matrices and vectors of conformable dimensions.

Let F denote the feasible set of INITIAL. F is the intersection of S and the polyhedron defined by the coupling constraints $Ax+Gy \leq b$. We assume that INITIAL has an optimal solution.

Note that implicit in this formulation is a partitioning of the constraint set into two subsets, one set defining the coupling constraints and the other set defining the polyhedron P. We take this partitioning as a given, although the results developed below can offer some guidance on how to select P. The main question addressed here is how to exploit the structure of INITIAL in order to solve large-scale instances effectively.

A basic concept underlying the inverse projection reformulation is the fact that any polyhedron can be generated from a finite set of extreme points and extreme rays. We need only consider extreme points because we assumed that P is bounded.

Let $proj_{\mathcal{B}}(S)$ denote the projection of S onto the subspace defined by the binary variables; that is, $proj_{\mathcal{B}}(S) := \{x \in \mathcal{B}^n : (x,y) \in S \text{ for some } y \in \mathcal{R}^p\}$. Now select any $X \subseteq \mathcal{B}^n$ such that $proj_{\mathcal{B}}(S) \subseteq X$. For each $x \in X$, define P_x to be the subset of the polyhedron P consisting of all elements whose binary subvector equals x; that is, $P_x := \{(u,v) \in \mathcal{B}^n \times \mathcal{R}^p : (u,v) \in P \text{ and } u = x\}$. Let E_x be the set of extreme points of P_x and let $E := \bigcup_{x \in X} E_x$ denote the union of all extreme points.

Let $C := [A \vdots G] \in \mathcal{R}^{m \times (n+p)}$ denote the concatenation of the A and G matrices. We associate with each element $x \in X$ a binary variable $\mu_x \in \mathcal{B}$. Additionally, we associate with each extreme point $e \in E_x$ a scalar variable $\lambda_{x,e} \in [0,1]$. We use the short-hand notation μ and λ to denote the vectors of these binary and scalar variables, respectively.

The *master formulation* (MASTER) corresponding to INITIAL is:

$$\max_{\mu,\lambda} \zeta(\lambda,\mu) := \sum_{x \in X} [\, f(x)\, \mu_x + \sum_{e=(x,y) \in E_x} l(y)\, \lambda_{x,e} \,] \tag{2.4}$$

subject to

$$\sum_{x \in X} \sum_{e \in E_x} (Ce)\, \lambda_{x,e} \leq b, \tag{2.5}$$

$$\mu_x = \sum_{e \in E_x} \lambda_{x,e}, \quad \forall x \in X, \tag{2.6}$$

$$\sum_{x \in X} \mu_x = 1, \tag{2.7}$$

$$\mu_x \in \mathcal{B}, \quad \forall x \in X, \tag{2.8}$$

$$\lambda_{x,e} \in \mathcal{R}_+, \quad \forall e \in E_x, \forall x \in X. \tag{2.9}$$

Note the following properties of the master formulation:

- A new set of decision variables replaces the original ones in INITIAL. The binary variable μ_x represents the selection of the integral portion of a solution. The set X can be viewed as the enumeration of all such integral solutions. The continuous portion of the solution is now represented as a convex combination of the extreme points of a polyhedron P_x as defined by the convexity weights $\{\lambda_{x,e}\}_{e \in E_x}$. The constraints $\mu_x = \sum_{e \in E_x} \lambda_{x,e}$, $\forall x \in X$ represent the 'activation' of the convexity constraint given the selection of integral component x.

- The objective function of MASTER is linear in the decision variables even though the original objective function of INITIAL is, in general, nonlinear in the binary variables. Therefore, MASTER is a linear mixed-binary program.

- The number of variables in MASTER is generally exponential in the number of variables in INITIAL. This arises from the enumeration of the discrete set X as well as the enumeration of all extreme points corresponding to P_x for each $x \in X$. The number of constraints in MASTER can also be higher than that of INITIAL if the cardinality of X is larger than the number of constraints required to define P.

- The validity of MASTER assumes the existence of at least one binary variable; that is $n \geq 1$. If INITIAL is a linear program with no integer variables, then we could perform a simple transformation where a dummy binary variable is augmented to each vector of continuous variables. If F is the feasible set of the original formulation then a new feasible set F' could be defined where $y \in F$ if and only if $(1, y) \in F'$. It is straightforward to observe that in this case constraints (2.7) and (2.8) can be eliminated from MASTER and the left hand-side of constraint (2.6) can be set to 1, yielding the traditional LP master formulation.

- If INITIAL is a pure binary program, then $E_x = \{x\}$ for each $x \in X$. In this case, constraint (2.7) represents a selection of an element of X.

The potential advantage of linearization is overshadowed by the explosion in number of variables. However, we show below that even if the objective function in INITIAL is linear, the master formulation typically has a stronger LP relaxation which, together with column generation techniques, can render it more tractable computationally. Before proving this in the next section, we need to define a third formulation that plays an intermediate role between INITIAL and MASTER. We refer to the following as the *convexified formulation* (CONV). It is defined within the same space of variables as the initial formulation.

$$\max_{x,y} \; z(x,y) := f(x) + l(y) \tag{2.10}$$

subject to

$$Ax + Gy \leq b, \tag{2.11}$$

$$(x,y) \in conv(E), \tag{2.12}$$

$$(x,y) \in (\mathcal{B}^n \times \mathcal{R}^p). \tag{2.13}$$

2.4 Analysis

In this section we analyze the relationship between INITIAL and CONV followed by the relationship between CONV and MASTER. Each of the three formulations discussed has a continuous relaxation version obtained simply by replacing the binary constraint $\mathcal{B}$ by the continuous unit interval $[0, 1]$. A summary of notation used is tabulated below.

Notation 2.1

		Formulation		
		INITIAL	*CONV*	*MASTER*
Binary	*Feasible set*	F	Q	Ω
version	*Optimal value*	z^{i*}	z^{c*}	z^{m*}
Continuous	*Feasible set*	$\overline{F}$	$\overline{Q}$	$\overline{\Omega}$
relaxation	*Optimal value*	$\overline{z}^{i*}$	$\overline{z}^{c*}$	$\overline{z}^{m*}$

2.4.1 Relation Between INITIAL and CONV

The result of this section can be summarized as follows: The binary version of the initial formulation is equivalent to the binary version of the convexified formulation. However, the continuous relaxation of the convexified formulation is at least as strong (and typically stronger) than that of the initial formulation. This is true as long as $proj_B(S) \subseteq X$.

Proposition 2.1

i) $\overline{F} \supseteq \overline{Q}$;

ii) $F = Q$;

iii) $z^{i*} = z^{c*}$;

iv) $\overline{z}^{i*} \geq \overline{z}^{c*}$.

Proof.

1. $\overline{F} \supseteq \overline{Q}$:

This follows from the fact that $E \subseteq P$ and therefore, $conv(E) \subseteq P$.

2. $F = Q$:

Note that by definition of E_x and the fact that P is bounded, $conv(E_x) = P_x$ for each $x \in X$. The integrality restrictions guarantee that the subvector of binary decision variables in any feasible solution equals some $x \in X$.

3. $z^{i*} = z^{c*}$:

This follows directly from (2) above because both INITIAL and CONV have the same objective function.

4. $\overline{z}^{i*} \geq \overline{z}^{c*}$:

This follows directly from (1) above because both INITIAL and CONV have the same objective function. ■

2.4.2 Relation Between CONV and MASTER

We now turn attention to the relation between the convexified and master formulations. The result here can be summarized as follows: The master and convexified formulations are equivalent (through a linear transformation) in their binary versions. In general, the continuous relaxation of MASTER is stronger than the continuous relaxation of CONV. The continuous relaxations become equivalent if the concave function f is affine. In this case, the convexified formulation can be viewed as the master formulation in the original variable space. This is true as long as $proj_{\mathcal{B}}(S) \subseteq X$.

Notation 2.2

For any point $(\lambda, \mu) \in \overline{\Omega}$ where λ is the vector $[\lambda_{x,e}]_{x \in X,\, e \in E_x}$ and μ is the vector $[\mu_x]_{x \in X}$, define the linear transformation

$$T((\lambda, \mu)) := \textstyle\sum_{x \in X} \sum_{e \in E_x} e\, \lambda_{x,e}. \tag{2.14}$$

For any set $U \subseteq \overline{\Omega}$, let

$$T(U) := \{T((\lambda, \mu)) : (\lambda, \mu) \in U\}.$$

In other words, $T(U)$ is the image of U under the linear transformation T defined by the extreme points $e \in E_x$, $x \in X$.

Proposition 2.2

i) $\overline{Q} = T(\overline{\Omega})$

ii) $Q = T(\Omega)$;

iii) $z^{c*} = z^{m*}$;

iv) $\overline{z}^{c*} \geq \overline{z}^{m*}$;

v) $\overline{z}^{c*} = \overline{z}^{m*}$ *if the function f is affine.*

Proof.

1. $\overline{Q} \subseteq T(\overline{\Omega})$:

Consider any $q = (x', y') \in \overline{Q}$. Therefore, $A\,x' + G\,y' \leq b$ and $q \in conv(E)$. The latter implies that there exists a vector $\lambda = [\lambda_{x,e}]_{x \in X,\, e \in E_x}$ of scalars satisfying

$$q = \sum_{x \in X} \sum_{e \in E_x} e\,\lambda_{x,e},$$
$$\sum_{x \in X} \sum_{e \in E_x} \lambda_{x,e} = 1, \text{ and}$$
$$\lambda_{x,e} \geq 0 \text{ for all } e \in E_x, x \in X.$$

Set the vector μ such that $\mu_x = \sum_{e \in E_x} \lambda_{x,e}\ \forall x \in X$. It can be verified that (λ, μ) satisfies the constraints of the continuous relaxation of MASTER and therefore $(\lambda, \mu) \in \overline{\Omega}$. This implies $q \in T(\overline{\Omega})$ by definition of the transformation T.

2. $\overline{Q} \supseteq T(\overline{\Omega})$:

Consider any $t = (x', y') \in T(\overline{\Omega})$. We need to show that t satisfies the constraints of the continuous relaxation of CONV. According to the definition of T, $t = \sum_{x \in X} \sum_{e \in E_x} e\,\lambda_{x,e}$ for some $(\lambda, \mu) \in \overline{\Omega}$. Therefore, (λ, μ) satisfies the constraints of the continuous relaxation of MASTER. The inequality $\sum\limits_{x \in X} \sum\limits_{e \in E_x} (C\,e)\,\lambda_{x,e} \leq b$, implies $A\,x' + G\,y' \leq b$. $(\lambda, \mu) \in \overline{\Omega}$ also implies that λ constitutes a valid vector of convexity weights. The relation $t = \sum_{x \in X} \sum_{e \in E_x} e\,\lambda_{x,e}$ implies $t \in conv(E)$. Finally it is easy to verify that $x' \in [0,1]^n$.

1 and 2 imply $\overline{Q} = T(\overline{\Omega})$.

3. $Q \subseteq T(\Omega)$:

The proof is essentially the same as that in (1) above. The only difference is that $q = (x', y') \in Q$ implies that $x' \in \mathcal{B}^n$. Therefore, the extreme points defining the convex combination of q must all have the same binary component x'. The expression $\mu_x = \sum_{e \in E_x} \lambda_{x,e}$ yields $\mu_x = 1$ if $x = x'$ and 0 otherwise. Therefore, μ is integral and $(\lambda, \mu) \in \Omega$.

4. $Q \supseteq T(\Omega)$:

The proof is the same as that in (2) above, with the exception that $(\lambda, \mu) \in \Omega$ implies that μ is integral and, therefore, $x' \in \mathcal{B}^n$.

3 and 4 imply $Q = T(\Omega)$.

5. For any $(\lambda, \mu) \in \Omega$ and $(x', y') = T((\lambda, \mu))$, $\zeta(\lambda, \mu) = z(x', y')$:

The binary restriction on μ in MASTER implies that $\mu_x = 1$ for $x = x'$ and 0 otherwise. Therefore,

$$\begin{aligned}
\zeta(\lambda, \mu) &= \sum_{x \in X} [\, f(x)\, \mu_x + \sum_{e=(x,y) \in E_x} l(y)\, \lambda_{x,e} \,] \\
&= f(x') + \sum_{e=(x',y) \in E_{x'}} l(y)\, \lambda_{x',e} \\
&= f(x') + \sum_{e=(x',y) \in E_{x'}} l(\lambda_{x',e}\, y) \\
&= f(x') + l(y') \\
&= z(x', y').
\end{aligned}$$

where we have utilized the assumption that the function l is affine in y and that $\sum_{e=(x',y) \in E_{x'}} \lambda_{x',e} = 1$.

6. $z^{c*} = z^{m*}$:

This follows directly from 3, 4, and 5 above.

7. For any $\omega = (\lambda, \mu) \in \overline{\Omega}$ and $(x', y') = T(\omega)$, $\zeta(\lambda, \mu) \leq z(x', y')$:

Note that the vector μ provides can be regarded as a vector of convexity weights applied to values of the function f.

$$\begin{aligned}
\zeta(\lambda, \mu) &= \sum_{x \in X} [\, f(x)\, \mu_x + \sum_{e=(x,y) \in E_x} l(y)\, \lambda_{x,e} \,] \\
&\leq f(x') + \sum_{x \in X} \sum_{e=(x,y) \in E_x} l(y)\, \lambda_{x,e} \\
&= f(x') + \sum_{x \in X} \sum_{e=(x,y) \in E_x} l(\lambda_{x,e} y) \\
&= f(x') + l(y') \\
&= z(x', y').
\end{aligned}$$

where the inequality stems from the assumption that f is concave. We have also utilized the assumption that the function l is linear in y.

8. $\overline{z}^{c*} \geq \overline{z}^{m*}$:

This follows directly from 1, 2, and 7 above.

9. $\overline{z}^{c*} = \overline{z}^{m*}$ if the function f is affine:

If the function is affine then the relation $\zeta(\lambda, \mu) \leq z(x', y')$ developed in 7 will hold with equality. ■

These results provide the main motivations behind using DW decomposition to solve large-scale mixed-binary problems: *the master formulation has a stronger continuous relaxation than the initial formulation.* This implies it is potentially more suited for LP-based branch-and-bound schemes.

2.4.3 Decomposable Systems

In various applications, the set S is decomposable into smaller independent subproblems. That is, $S = S^1 \times S^2 \times \cdots \times S^J$, typically corresponding to a block diagonal structure of the constraint matrix describing the polyhedron P. If E^j denotes the set of finite generators of S^j then it can be shown that the set of finite generators of the full set S can be expressed as $E = E^1 \times E^2 \times \cdots \times E^J$. In this case the master formulation assumes the form below.

Here we have employed a straightforward extension of the notation introduced earlier, with superscripts denoting subproblems. We have also partitioned the matrix $C := [A \vdots G]$ into submatrices $C^j := [A^j \vdots G^j]$ corresponding to the variables in each of the independent subproblems.

$$\max_{\mu,\lambda} \zeta(\lambda, \mu) := \sum_{j=1}^{J} \sum_{x \in X} [\, f(x)\, \mu_x^j + \sum_{e=(x,y) \in E_x^j} l(y)\, \lambda_{x,e}^j \,] \tag{2.15}$$

subject to

$$\sum_{j=1}^{J} \sum_{x \in X^j} \sum_{e \in E_x^j} (C^j\, e)\, \lambda_{x,e}^j \leq b, \tag{2.16}$$

$$\mu_x^j = \sum_{e \in E_x^j} \lambda_{x,e}^j, \quad \forall x \in X^j,\ j = 1, ..., J, \tag{2.17}$$

$$\sum_{x \in X^j} \mu_x^j = 1,\ j = 1, ..., J, \tag{2.18}$$

$$\mu_x^j \in \mathcal{B}, \quad \forall x \in X,\ j = 1, ..., J, \tag{2.19}$$

$$\lambda_{x,e}^j \in \mathcal{R}_+, \quad \forall e \in E_x^j, \forall x \in X^j,\ j = 1, ..., J. \tag{2.20}$$

The presence of a decomposable structure, especially one where $|E^j|$ is small for each j, typically tames the exponential explosion of the number of variables in the master formulation. The strive to exploit such decomposition provided the initial motivation for DW reformulations and explains why it is often called DW *decomposition.*

2.5 Solution Algorithms

2.5.1 Solving the Linear Programming Relaxation of the Master Formulation

The LP relaxation of the master formulation has more than $|E|$ variables. As $|E|$ is typically huge, it is impractical to solve this LP directly. Instead, delayed column generation methods are used. The standard implementation of column generation for (pure continuous) linear programs is a well established technique and the reader is referred to any of the standard textbooks covering it such as [BeT97]. Briefly stated, a restricted master problem is solved on a subset of variables and additional variables (elements of E) are identified that could improve the objective value through a pricing subproblem. The columns are generated and added to the restricted master problem. The process is repeated until no more columns can be generated and the procedure terminates with an optimal solution to the master. Column generation can be viewed as an implementation of the revised-simplex method with Dantzig's pivoting rule where entering variables are generated by the pricing subproblem.

The objective function of the pricing subproblem is derived from the optimal dual variables associated with the restricted master problem. The extreme points of the feasible set of the pricing subproblem are the incidence vectors of elements of E. Hence the pricing problem is a mixed-binary program.

2.5.2 Solving the Master Formulation to Integrality

In the case of mixed-integer programs, the master formulation needs to be solved to integrality. An LP-based branch-and-bound scheme is typically employed where the LP at each node is again solved using LP column generation. The approach is called branch-and-price. The implementation is not straightforward because variable fixing has implications on the structure of the subproblem. For an exposition of the issues associated with branch-and-price, the reader is referred to Barnhart et al. [BJN98].

Chapter 3

A Class of Weakly-Coupled Network Design Problems

In this chapter we specialize the DW reformulation strategy, developed in Chapter 2, to a class of 'weakly-coupled' network design problems. This class of problems is a generalization of the airline fleet assignment application discussed in Chapter 4.

We begin by stating a standard form of the network design problem. We then introduce a number of modifications that eventually lead to a variation of the standard problem that we label as 'weakly-coupled in the *flow* variables'. This is essentially a problem with path restrictions on commodity flows but with 'soft' demand requirements. We prove that the decision version of this problem is NP-complete. We then investigate a solution approach based on DW reformulation tailored to instances of this problem including those that are 'strongly-coupled in the *design* variables'. The procedure is designed to achieve a trade-off between accuracy and tractability.

3.1 Problem Statement

Consider a simple directed graph $G = (N, A)$ where N is the set of nodes and A is the set of arcs. Associated with each arc $a \in A$ is a finite set $C_a \subseteq \mathcal{Z}_+$ of *capacity levels.* We note that a capacity level set C_a might be specified more compactly than by a list of values. For instance, it could be specified as all non-negative integer combinations of a small number of levels. The cost of providing capacity on arc a is given by a nondecreasing function $\gamma_a : C_a \rightarrow \mathcal{Z}$. We assume that exactly one capacity level must be chosen for each arc. The set C_a may of course include a 'zero' capacity level.

The network is required to support the flows of a set K of commodities. We assume, with some loss of generality, that each commodity k is specified by a single origin-destination pair. Note, however, that in many significant applications, such as telecommunication networks, this assumption is natural. For each commodity k, the origin node is denoted $o(k)$ and the destination node is denoted $d(k)$. Let $\mathcal{P}^k$ denote the set of all directed paths from $o(k)$ to $d(k)$ in G. A flow cost $\alpha_p^k \in \mathcal{Z}_+$ is associated with each unit of flow of commodity k on path $p \in \mathcal{P}^k$. We initially assume that a fixed demand of $b^k \in \mathcal{Z}_+$ units is required to flow from origin to destination.

The decision variable f_p^k represents the flow quantity of commodity k on path $p \in \mathcal{P}^k$. The design variable x_a represents the capacity level selected on arc a. The short-hand notation f and x will be used to denote the vectors of all flow variables and all design variables, respectively.

The *standard network design problem* (NDP) can be formulated as follows:

$$z^*_{NDP} = \min_{x,f} \sum_{a \in A} \gamma_a(x_a) + \sum_{k \in K} \sum_{p \in \mathcal{P}^k} \alpha_p^k f_p^k \tag{3.1}$$

subject to

$$\sum_{k \in K} \sum_{p \in \mathcal{P}^k : p \ni a} f_p^k - x_a \leq 0, \qquad a \in A, \tag{3.2}$$

$$\sum_{p \in \mathcal{P}^k} f_p^k = b^k, \qquad k \in K, \tag{3.3}$$

$$f_p^k \in \mathcal{R}_+, \qquad p \in \mathcal{P}^k,\ k \in K, \tag{3.4}$$

$$x_a \in C_a, \qquad a \in A. \tag{3.5}$$

Essentially the problem is to determine where to install capacity and how to allocate flow among alternative paths in order to minimize total cost.

Problem NDP and a number of its special cases are NP-hard. Also, these problems often exhibit weak LP relaxations. For surveys on applications, polyhedral structure, and solution algorithms of NDP the reader is referred to Balakrishnan, Magnanti, and Mirchandani [BMM95a], Magnanti and Wong [MaW84], and Minoux [Min89].

We now define a variant of $[NDP]$ by introducing two modifications:

1. We allow 'soft' demand; that is, a demand *range* $[\underline{b}^k, \bar{b}^k]$, $0 \leq \underline{b}^k \leq \bar{b}^k$, is specified for each commodity k. Any flow from origin to destination whose total value for each commodity falls within the specified range is considered to have met demand requirements.

2. We restrict the set of path flows for each commodity to a single path only. That is, we replace $\mathcal{P}^k$ by a single path $P^k \in \mathcal{P}^k$ that any flow of commodity k must follow. We allow the possibility that flow along a path might incur a per unit revenue as well as a per unit cost. We choose a revenue viewpoint, and associate with each unit of flow of commodity k along path P^k a 'net revenue' $r^k \in \mathcal{Z}$, which can be positive, zero, or negative.

For simplicity, we make two assumptions that are non-restrictive:

A1. $r^k \geq 0$: Instances with commodities having negative r^k values can be pre-processed by setting the flow values for those commodities equal to their minimum demand requirements and updating capacity levels accordingly.

A2. $\underline{b}^k = 0$: Any instance can be reduced to this form through a simple additive transformation of the variables.

Letting f^k denote the flow variable associated with commodity k, the modified problem, which we refer to as the *weakly-coupled network design problem* (WNDP), is formulated as follows:

$$z^*_{WNDP} = \max_{x,f} z_{WNDP}(x,f) := \sum_{k\in K} r^k f^k - \sum_{a\in A} \gamma_a(x_a) \tag{3.6}$$

subject to

$$\sum_{k\in K:a\in P^k} f^k - x_a \le 0, \qquad a \in A, \tag{3.7}$$

$$0 \le f^k \le \bar{b}^k, \qquad k \in K, \tag{3.8}$$

$$x_a \in C_a, \qquad a \in A. \tag{3.9}$$

The assumption of single predefined paths for each commodity is clearly restrictive. However, it does capture a number of applications of interest. In particular, the airline fleet assignment application discussed in Chapter 4 satisfies this assumption where a commodity represents an itinerary which is a predefined sequence of flight legs (arcs). In some simple hub-and-spoke networks, the assumption of unique paths might also be justifiable. More importantly, however, the single path assumption is a first step towards understanding network structures characterized by a small number of alternative paths.

The single path assumption does away with routing decisions. The main trade-off is between revenue gained from quantity supplied, on the one hand, and capacity cost paid to support such flow, on the other hand. Surprisingly, the author is not aware of any systematic body of literature that addresses the computational issues surrounding WNDP.

3.2 Computational Complexity

Before addressing the complexity of WNDP, we draw attention in the next example to a polynomially solvable special case.

Example 3.1

Consider an instance of WNDP where commodity paths do not intersect. That is, for any two distinct commodities k_1 *and* k_2, $P^{k_1} \cap P^{k_1} = \phi$. *In this case, the problem decomposes by commodity. Therefore, assume that the underlying graph* $G = (N, A)$ *consists of a single path* $P = (v_1, v_2, ..., v_n)$ *where* $N = \{v_i : i = 1, ..., n\}$ *and* $A = \{(v_i, v_{i+1}) : i = 1, ..., n-1\}$.

Assume C_a is specified by a list of values for each $a \in A$ (as opposed to succinct description giving rise to a super-polynomial set of capacity levels). K consists of a single commodity with origin v_1, destination v_n, per unit revenue $r \geq 0$, and demand limits $\underline{b}$ and $\bar{b}$. Let f denote the commodity's flow value.

This simple problem can be solved as follows. Let $C = \bigcup_{a \in A} C_a$. An optimal solution exists where $f \in C$. Therefore, for each $c \in C$, set $x_a(c) = \min\{l \in C_a : l \geq c\}$. Compute $z(c) := r\,c - \sum_{a \in A} \gamma_a(x_a(c))$. Finally, pick the solution yielding the maximum value of $z(c)$.

The question we wish to address now is: what is the computational complexity of WNDP? More precisely, what is the computational complexity of the following decision version of WNDP:

Problem 3.1 $WNDP'$

INSTANCE: Simple directed graph $G = (N, A)$; commodity set K, each commodity characterized by an origin node, $o(k)$, a destination node, $d(k)$, a path $P^k \in \mathcal{P}^k$, an upper demand requirement, $\bar{b}^k \in \mathcal{Z}_+$, and a per unit revenue, $r^k \in \mathcal{Z}$; capacity level set $C_a \subseteq \mathcal{Z}_+$ and capacity cost function $\gamma_a : C_a \to \mathcal{Z}$ for each $a \in A$; a scalar $s \in \mathcal{Z}$.

QUESTION: Does there exist a feasible solution (x, f) to $WNDP$ with $z_{WNDP}(x, f) \geq s$?

Proposition 3.1

Problem $WNDP'$ is NP-complete.

Proof.

Clearly, $WNDP'$ is in the class NP. We show it is NP-complete by reduction from INDEPENDENT SET (problem [*GT*20] in Garey and Johnson [GaJ79]).

An instance I' of INDEPENDENT SET is given by an undirected graph $G' = (V, E)$ and a positive integer $s \leq |V|$. The question is whether G' contains an independent set of cardinality s or more. $X \subseteq V$ is an independent set if and only if no two vertices in X share an edge in E. Let $n := |V|$ and $m := |E|$. Label the vertices and edges of G' such that $V = \{v_1, ..., v_n\}$ and $E = \{e_1, ..., e_m\}$. Let E_i denote the set of edges incident on vertex v_i.

We polynomially transform I' into an instance I of $WNDP'$ on a simple directed graph $G = (N, A)$. The idea of the transformation is to associate with each vertex in V a commodity in G. Paths for two commodities are constructed so that they intersect if and only if the corresponding vertices share an edge. The remaining problem parameters are set such that $WNDP'$ reduces to the problem of selecting the maximum number of arc-disjoint paths. An illustration of this transformation is provided at the end of the proof. Formally, the instance of $WNDP'$ is constructed as follows:

1. $N := \{n_{i,j} : i = 0, ..., m \text{ and } j = 0, ..., n\}$;
2. $A = A^1 \cup A^2$ where
   - $A^1 := \{(n_{i,j}, n_{i+1,j}) : i = 0, ..., m-1;\ j = 0, ..., n\}$; and
   - $A^2 := \{(n_{i,j}, n_{i,0}) : i = 0, ..., m-1;\ j = 1, ..., n\} \cup \{(n_{i,0}, n_{i,j}) : i = 1, ..., m;\ j = 1, ..., n\}$.
3. $K = \{1, ..., n\}$. For each $k \in K$,
   - $o(k) = n_{0,k}$;
   - $d(k) = n_{m,k}$;
   - P^k is the unique path defined by the union of the following sets of arcs:
     - $\{(n_{i-1,0}, n_{i,0}) : e_i \in E_k\} \subseteq A^1$;
     - $\{(n_{i-1,k}, n_{i,k}) : e_i \notin E_k\} \subseteq A^1$;
     - $\{(n_{0,k}, n_{0,0}) : e_1 \in E_k\} \cup \{(n_{i,k}, n_{i,0}) : e_i \notin E_k \text{ and } e_{i+1} \in E_k \text{ for } 1 \le i \ge m-1\} \subseteq A^2$;
     - $\{(n_{i,,0}, n_{i,k}) : e_i \in E_k \text{ and } e_{i+1} \notin E_k \text{ for } 1 \le i \le m-1\} \cup \{(n_{m,0}, n_{m,k}) : e_m \in E_k\} \subseteq A^2$;
   - $\bar{b}^k = 1$;
   - $r^k = m + 1$;
4. $C_a = \{0, 1\}$ for all $a \in A$;

5. $\gamma_a(0) = 0$ for all $a \in A$; $\gamma_a(1) = 1$ for all $a \in A^1$ and 0, otherwise.

This transformation is polynomial in the size of an INDEPENDENT SET instance. It is useful to think of the each arc $(n_{i-1,0}, n_{i,0})$ in G as associated with edge e_i in G' and each commodity k in I as associated with vertex v_k in G'. Note that if paths P^{k_1} and P^{k_2} intersect then their intersection is precisely the arc corresponding to the edge $\{v_{k_1}, v_{k_2}\}$ in G'. Also note that $|P^k \cap A^1| = m$ for all $k \in K$.

We now show that there exists an independent set, X, in instance I' of cardinality $|X| \geq s$ if and only if there exists a feasible solution $(\mathbf{x}, \mathbf{f})$ to $WNDP$ of value $z_{WNDP}(\mathbf{x}, \mathbf{f}) \geq s$.

i) Suppose an independent set, X, exists of cardinality $|X| \geq s$. Then construct a feasible solution to $WNDP$ by selecting capacity level 1 for each arc on each path corresponding to a vertex in the independent set. That is, $C_a = 1$ for all $a \in \{a' \in P^k : v_k \in X\}$. All other capacity levels can be set to zero. Note that, as observed above, the paths whose capacity equal 1 do not intersect. Therefore, each path can carry a total flow of value 1. The cost of each open path is m while the revenue gained is $m+1$. Because the number of open paths is greater than or equal to s therefore the feasible solution to $WNDP$ has value $\geq s[(m+1) - m] = s$.

ii) Now consider a feasible solution $(\mathbf{x}, \mathbf{f})$ to $WNDP$ of value $z_{WNDP}(\mathbf{x}, \mathbf{f}) \geq s$. We can assume that any commodity k whose flow value is positive must have a flow value of 1. To see this note that each arc in P^k must have capacity level 1 in order to carry this flow. If flow value cannot increase to its upper demand limit $\overline{b}^k = 1$ because of another commodity's flow sharing some arc in P^k then we can always decrease the other commodity's flow and increase the flow of commodity k without deteriorating the objective function. In fact, when the other commodity's drops to zero, we can 'close' its path thus strictly improving the objective function value. Under this assumption all paths are disjoint because any arc can only carry a maximum flow of 1. This implies that the corresponding vertices in G' form an independence set X. The number of open paths is greater than or equal to s because the net revenue of a single path flow is $(m+1) - m = 1$. Therefore, $|X| \geq s$.

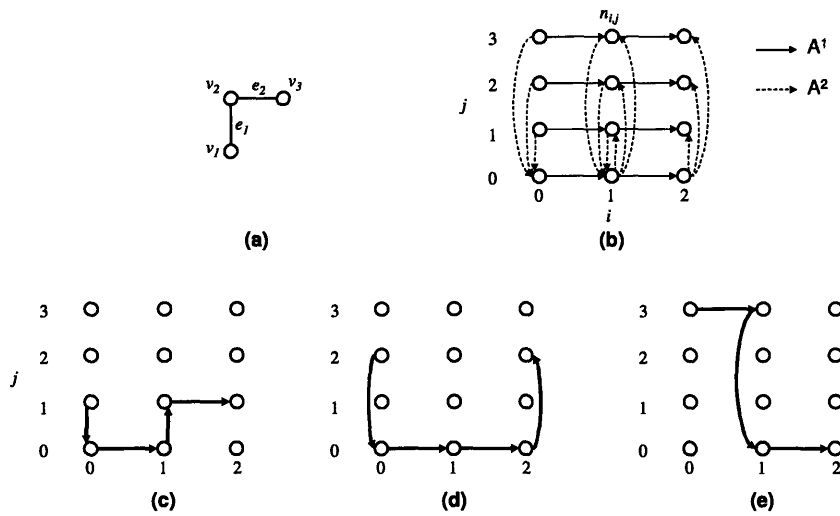

Figure 3-1: Transformation used in proving NP-completeness of $WNDP'$.

Therefore, a polynomial time algorithm to solve $WNDP'$ yields a polynomial time algorithm to solve INDEPENDENT SET. The latter is NP-complete and therefore $WNDP'$ is NP-complete.

Example 3.2 *We illustrate the polynomial transformation employed in the proof through an example based on an instance of INDEPENDENT SET with G' consisting of three vertices and two edges. A sketch of G' is provided in figure 3-1 (a). The corresponding transformation G is drawn in figure 3-1 (b). The paths corresponding to vertices v_1, v_2, and v_3 are highlighted in figures 3-1 (c), (d), and (e), respectively.*

■

It should be pointed out that the reduction presented in the proof of Proposition 3.1 is *approximation preserving*[1]. This implies that a constant factor polynomial approximation algorithm for $WNDP$ yields a constant factor polynomial approximation algorithm for INDEPENDENT SET. The latter is known to be *inapproximable* within any constant

[1] I thank Professor Andreas Schulz for pointing out this fact.

factor, assuming $P \neq NP$. Therefore, $WNDP$ is also *inapproximable* within any constant factor, assuming $P \neq NP$.

3.3 WNDP with Coupling Capacity Constraints

The result of Proposition 3.1 motivates the study of algorithms that solve WNDP approximately. Before introducing such algorithms, we first expand the scope of WNDP by allowing the presence of a set of linear inequalities $D\mathbf{x} \leq \mathbf{d}$ that couple the design variables. These inequalities might represent budget limitations, survivability requirements, or other operational constraints. We arrive at the *extended weakly-coupled network design problem* WNDP+:

$$z^*_{WNDP+} = \max_{x,f} z_{WNDP+}(x, f) := \sum_{k \in K} r^k f^k - \sum_{a \in A} \gamma_a(x_a) \tag{3.10}$$

subject to

$$D\,x \leq d, \tag{3.11}$$

$$\sum_{k \in K : a \in P^k} f^k - x_a \leq 0, \qquad a \in A, \tag{3.12}$$

$$0 \leq f^k \leq \bar{b}^k, \qquad k \in K, \tag{3.13}$$

$$x_a \in C_a, \qquad a \in A. \tag{3.14}$$

WNDP+ is the main problem addressed in the rest of this chapter. The single path flow assumption suggests that it is weakly coupled in the flow variables. However, the design variables may be strongly coupled. Our observation is that in many practical instances the polyhedron defined by $D\,x \leq d$ exhibits low fractionality. The introduction of flow variables in the formulation is what generates excessive fractionality. WNDP+ can be strengthened by applying DW reformulation using the set defined by constraints (3.12) - (3.14) as a subproblem. A straight-forward implementation of DW-reformulation results in a large number of variables and a pricing problem that is not necessarily easy to solve. The next section introduces a bounded-approximation step that combined with DW reformulation yields a tractable procedure for solving WNDP+.

3.4 Approximation Step

To avoid a huge master formulation whose variables correspond to all possible capacity assignments to the entire network, we explore an approach that *induces* a block angular structure of WNDP+. Put differently, we examine an approach that decomposes the entire network into subnetworks with a small amount of inter-subnetwork flow. This necessitates defining and comparing different *partitions* of the set of arcs.

3.4.1 Partitions

Definition 3.1 (Partitions)

A partition Π *of a set of arcs* A *is a collection of mutually exclusive and collectively exhaustive subsets of* A*; that is,*

$$\Pi = \left\{ A^i \subseteq A : \bigcup_{i=1}^{\pi} A^i = A,\ A^i \neq \phi,\ \forall i,\ \text{and}\ A^u \cap A^v = \phi,\ \forall u \neq v \right\},$$

where π *is the cardinality of the collection. Each element of the partition is referred to as a 'subnetwork'.*

The minimal partition, $\underline{\Pi}$*, is the special partition* $\underline{\Pi} := \{\{a\} : a \in A\}$.

The maximal partition, $\overline{\Pi}$*, is the special partition* $\overline{\Pi} := \{A\}$.

It is necessary for analytical and computational purposes to compare the *degree of consolidation* of different partitions. This can be done using the concept of *nesting*, defined next, which is a standard concept in various combinatorial applications.

Definition 3.2 (Nesting)

Partition Π *is nested in partition* Π' *(denoted* $\Pi \preceq \Pi'$*) if every member of* Π *is a subset of some member of* Π'*. We say that* Π *is finer than* Π'.

Clearly, $\underline{\Pi} \preceq \Pi \preceq \overline{\Pi}$ for any partition Π of A.

We next define a computationally important partition.

Definition 3.3 (Complete Partition)

The complete partition, Π^c*, is a minimal partition (with respect to nesting) having no commodity paths crossing from one subnetwork to another. Formally,* Π^c *is defined by the following algorithm:*

Algorithm 3.1 COMPLETE_PARTITION:

1. *For each arc* $a \in A$*, define a node* n_a*. For each commodity* $k \in K$*, define a node* n_k*;*

2. *Set* $L := \{n_a : a \in A\}$*;* $R := \{n_k : k \in K\}$*;*

3. *For each* $k \in K$ *and* $a \in P^k$*, define an edge* $\{n_a, n_k\}$*. Let* H *denote the set of all such edges;*

4. *Define the bipartite graph* $G' := (L \cup R, H)$*;*

5. *Decompose* G' *into maximally connected components;*

6. *Set* Π^c *to be the partition of* L *defined by the decomposition performed in Step 5.*

Intuitively, Π^c, is the finest partition of *flow-independent* subnetworks. The size of the subnetworks produced is a rough indication of the degree of flow coupling in the network. It is more desirable from a computational point of view for Π^c to consist of many small sized subnetworks than of a few large ones.

The choice of partition significantly impacts the degree of approximation and the computational performance of the formulation developed below. The finer the partition, the fewer the columns in a master formulation, but also the larger the approximation error due to a greater amount of inter-subnetwork flow. We defer discussion of how to select the best partition that achieves a required trade-off between accuracy and tractability till the end of this section. For current purposes, we assume that an arbitrary partition $\Pi = \{A^1, A^2, ..., A^\pi\}$ has been pre-selected.

3.4.2 Revenue Allocation

We next outline a simple procedure for allocating the per unit path revenues, r^k, of each commodity k to the arcs in P^k. This is a necessary step for decomposing the network into 'revenue-independent' subnetworks.

Definition 3.4 (Revenue allocation schemes)

i) A revenue allocation scheme is a set of allocation weights $\{\rho_a^k : k \in K \text{ and } a \in A\}$.

ii) An allocation scheme is valid if the allocation weights satisfy the following properties:

- $\rho_a^k \geq 0$ *for all* $k \in K$ *and* $a \in A$;
- $\rho_a^k = 0$ *if* $a \notin P^k$, *for all* $k \in K$; *and*
- $\sum_{a \in A} \rho_a^k = 1$, *for all* $k \in K$.

Given a valid allocation scheme $\{\rho_a^k\}_{a \in P^k, k \in K}$ and a partition $\Pi = \{A^1, ..., A^\pi\}$ of A, we define allocation weights to *subnetworks* as follows:

$$\rho^{k,i} = \sum_{a \in A^i} \rho_a^k, \qquad k \in K,\ i = 1, ..., \pi. \tag{3.15}$$

This allocation scheme is used to define 'self-contained' revenues for each subnetwork in a partition.

3.4.3 Decomposition

Let K^i denote the set of commodities whose paths include an arc in A^i; that is, $K^i := \{k \in K : P^k \cap A^i \neq \phi\}$. Let x^i denote the subvector of the design vector x corresponding to the arcs in A^i. Problem WNDP+ can be rewritten as follows:

$$z^*_{WNDP+} = \max_{x,f} z_{WNDP+}(x, f) := \sum_{i=1}^{\pi} \sum_{k \in K^i} \rho^{k,i} r^k f^{k,i} - \sum_{i=1}^{\pi} \sum_{a \in A^i} \gamma_a(x_a) \tag{3.16}$$

subject to

$$\sum_{i=1}^{\pi} D^i x^i \leq d, \tag{3.17}$$

$$\sum_{k \in K^i : a \in P^k} f^{k,i} - x_a \leq 0, \qquad a \in A^i,\ i = 1, ..., \pi, \tag{3.18}$$

$$f^{k,i} = f^k, \qquad k \in K^i,\ i = 1, ..., \pi, \tag{3.19}$$

$$0 \leq f^{k,i} \leq \bar{b}^k, \qquad k \in K^i,\ i = 1, ..., \pi, \tag{3.20}$$

$$x_a \in C_a, \qquad a \in A^i,\ i = 1, ..., \pi. \tag{3.21}$$

In the above formulation, the flow decision variable corresponding to each commodity has been replaced by multiple variables, one for each subnetwork of Π having an arc in the commodity's unique path. The variables $f^{k,i}$ can be viewed as 'local decisions' made by subnetwork A^i as to how much flow of commodity k should the entire network support. However, the per unit revenue associated with that decision is $\rho^{k,i} r^k$, not r^k. *Consistency* among these local decisions is imposed by constraints (3.19).

It can be seen that subproblem (3.18) - (3.21) decomposes by subnetwork if the flow consistency constraints, (3.19), are relaxed. This suggests dualizing them through a Lagrange multiplier approach. Let $\lambda^{k,i}$ denote the Lagrange multiplier associated with commodity k and subnetwork A^i and let λ denote the vector of all these multipliers. The dual function $L(\lambda)$ is the optimal value of the following problem which we refer to as WNDP+(Π) to emphasize its dependance on the selection of partition Π:

$$L(\lambda) = \max_{x,f} \sum_{i=1}^{\pi} \sum_{k \in K^i} (\rho^{k,i} r^k - \lambda^{k,i}) f^{k,i} - \sum_{i=1}^{\pi} \sum_{a \in A^i} \gamma_a(x_a) + \sum_{k \in K} (\sum_{i: k \in K^i} \lambda^{k,i}) f^k \tag{3.22}$$

subject to

$$\sum_{i=1}^{\pi} D^i x^i \leq d, \tag{3.23}$$

$$\sum_{k \in K^i : a \in P^k} f^{k,i} - x_a \leq 0, \qquad a \in A^i,\ i = 1, ..., \pi, \tag{3.24}$$

$$0 \leq f^{k,i} \leq \bar{b}^k, \qquad k \in K^i,\ i = 1, ..., \pi, \tag{3.25}$$

$$x_a \in C_a, \qquad a \in A^i,\ i = 1, ..., \pi. \tag{3.26}$$

The block angular structure of WNDP+(Π) suggests that it might be easier to solve than WNDP+ by employing DW reformulation. The next section analyzes the properties of WNDP+(Π), including a bound on its degree of approximation as a function of Π.

3.4.4 Analysis

The following two properties of the dual function in WNDP+(Π) are important to note. For proofs, the reader is referred to any standard treatment of Lagrangian duality (for instance, [Wol98]).

Proposition 3.2

1. *$L(\lambda) \geq z^*_{WNDP+}$ for any choice of Lagrange multipliers λ;*

2. *$L(\lambda)$ is a convex function of λ.*

Therefore each choice of Lagrange multipliers yields a dual function value that is an upper bound on z^*_{WNDP+}. The dual problem DUAL(Π) is the problem of finding the smallest upper bound:

$$z^*_D := \min_{\lambda} L(\lambda). \tag{3.27}$$

Proposition 3.3

There exists an optimal solution λ to the dual problem DUAL(Π) satisfying:

1. *$\sum_{i:k\in K^i} \lambda^{k,i} = 0 \quad \forall k \in K$; and*

2. *$\rho^{k,i} r^k - \lambda^{k,i} \geq 0 \quad \forall k \in K$ and i such that $k \in K^i$.*

Proof.

1. Consider problem WNDP+(Π) defining the dual function. If $\sum_{i:k\in K^i} \lambda^{k,i} > 0$ for some $k \in K$ then the optimal value can be made arbitrarily large by setting f^k to an arbitrarily large positive value. Similarly, if $\sum_{i:k\in K^i} \lambda^{k,i} < 0$ for some $k \in K$ then the optimal value can be made arbitrarily large by setting f^k to an arbitrarily small negative value. Because the dual problem has a finite optimal value, therefore an optimal solution must have $\sum_{i:k\in K^i} \lambda^{k,i} = 0$ for all $k \in K$.

2. Property (1) of the proposition, the definition of valid allocation schemes, and the assumption that $r^k \geq 0$ ensure that for each $k \in K$,

$$\begin{aligned}
\textstyle\sum_{i:k\in K^i} (\rho^{k,i_1} r^k - \lambda^{k,i_1}) &= \textstyle\sum_{i:k\in K^i} \rho^{k,i_1} r^k - \sum_{i:k\in K^i} \lambda^{k,i_1} \\
&= \textstyle\sum_{i:k\in K^i} \rho^{k,i_1} r^k \\
&= r^k \\
&\geq 0
\end{aligned}$$

Suppose λ is a solution to the dual problem satisfying property (1) of the proposition, but having $\rho^{k,i_1}r^k - \lambda^{k,i_1} < 0$ for some commodity k and subnetwork i_1. The inequality above implies that there exists another subnetwork i_2 with $\rho^{k,i_2}r^k - \lambda^{k,i_2} > 0$. In an optimal solution to WNDP+(II), $f^{k,i_1} = 0$ while $f^{k,i_2} \geq 0$. By reducing the value of λ^{k,i_1} by a finite amount ϵ, we can increase the value of λ^{k,i_2} by the same amount maintaining property (1) of the proposition and the non-negativity of $\rho^{k,i_2}r^k - \lambda^{k,i_2}$. We now claim that the optimal value of problem WNDP+(II) with this modified objective function cannot increase. To see this, note that after fixing $f^{k,i_1} = 0$, we have an optimization problem with the same feasible set of nonnegative flows but with an objective function whose coefficients are strictly smaller. Repeating this process for each term $\rho^{k,i}r^k - \lambda^{k,i} < 0$ leads to an alternative optimal solution to the dual problem satisfying property (2) of the proposition.

■

Proposition 3.3 and the definition of a valid revenue allocation scheme imply that for each commodity k, $\sum_{i=1}^{\pi}(\rho^{k,i}r^k - \lambda^{k,i}) = r^k$ at some optimal solution to the dual problem. Therefore one interpretation of the application of Lagrangian relaxation to problem WNDP+ is *re-allocation of commodity revenues to the subnetworks forming* Π, *always ensuring that, for each commodity, the revenue allocated to each subnetwork is non-negative and that the sum of allocations equals the original revenue.* A valid revenue allocation scheme as defined in Section 3.4.2 can, therefore, be viewed as an initial estimate of Lagrange multipliers. The objective of the dual problem is to find an allocation that leads to the lowest upper bound. Proposition 3.2 implies $z_D^* \geq z_{WNDP+}^*$. As the next example shows, equality need not hold.

Example 3.3

Consider an instance of WNDP+ defined on a simple graph consisting of two arcs (see Figure 3-2). There is a single commodity, k, with $\bar{b}^k = 100$, *and* $r^k = 1$. $P^k = (a_1, a_2)$. *Each arc has two capacity levels. Arc* a_1 *can be assigned a capacity level of* 0 *at no cost or a capacity level of* 10 *at a cost of* 5. *Arc* a_2 *can be assigned a capacity level of* 0 *at no cost*

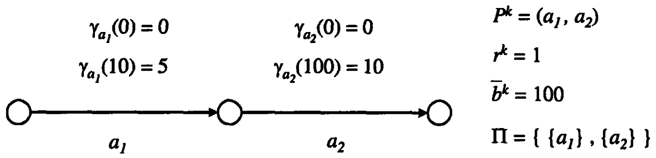

Figure 3-2: Example showing duality gap for Problem DUAL(Π).

or a capacity level of 100 *at a cost of* 10. *Assume the coupling constraints* $\sum_{i=1}^{\pi} D^i \mathbf{x}^i \leq \mathbf{d}$ *are non-binding. By inspection, it can be verified that* $z^*_{WNDP+} = 0$.

Now consider the partition $\{A^1, A^2\}$ *where* $A^1 = \{a_1\}$ *and* $A^2 = \{a_2\}$. *Let* θ_1 *and* θ_2 *denote, respectively, the coefficients of* $f^{k,1}$ *and* $f^{k,2}$ *in (3.22). We have established above that an optimal solution exists with* $\theta_1 + \theta_2 = r^k = 1$. *Consider the following three collectively exhaustive cases:*

case 1: $\theta_1 \leq 0.5$ *and* $\theta_2 = 1 - \theta_1 \geq 0.5$. *Here subnetwork* A^1 *selects* $x_{a_1} = 0$ *and* $f^{k,1} = 0$ *and subnetwork* A^2 *selects* $x_{a_2} = 100$ *and* $f^{k,2} = 100$. $L(\lambda) = 100\,\theta_2 - 10 > 0$.

case 2: $0.5 < \theta_1 \leq 0.9$ *and* $0.1 \leq \theta_2 < 0.5$. *Here subnetwork* A^1 *selects* $x_{a_1} = 10$ *and* $f^{k,1} = 10$ *and subnetwork* A^2 *selects* $x_{a_2} = 100$ *and* $f^{k,2} = 100$. $L(\lambda) = (10\,\theta_1 - 5) + (100\,\theta_2 - 10) = 85 - 90\,\theta_1 > 0$.

case 3: $0.9 < \theta_1$ *and* $\theta_2 = 1 - \theta_1 < 0.1$. *Here subnetwork* A^1 *selects* $x_{a_1} = 10$ *and* $f^{k,1} = 10$ *and subnetwork* A^2 *selects* $x_{a_2} = 0$ *and* $f^{k,2} = 0$. $L(\lambda) = 10\,\theta_1 - 5 > 0$.

Therefore, $z^*_D > 0 = z^*_{WNDP+}$.

To complement the upper bounds provided by the dual function, we need lower bounds on z^*_{WNDP+}. These lower bounds correspond to feasible solutions whose deviation from optimality can be assessed using the Lagrangian upper bounds. Fortunately, simple lower bounds are readily computable. This is possible because of the fact that any capacity configuration obtained by solving WNDP+(Π) has a corresponding feasible flow solution. To formalize this observation, we consider three different lower bounds that are successively tighter but more computationally intensive. Assume that we are given:

i) a partition $\Pi = \{A^1, A^2, ..., A^\pi\}$ of the set of arcs A; and

ii) a set of capacity levels $\{\widehat{x}_a\}_{a\in A}$ together with a set of local flow decisions $\{\widehat{f}^{k,i}\}_{k\in K^i, i=1,...,\pi}$ jointly satisfying constraints (3.23) - (3.26). These values could be a solution of WNDP+(II) for some vector λ of Lagrange multipliers.

Let

$$\widehat{f}^k := \min_{i:k\in K^i} \widehat{f}^{k,i}, k \in K.$$

Therefore, $\widehat{f}^k$ is a conservative flow value taken to be the minimum of all local subnetwork flows. Let K' denote the set of commodities whose paths span more than one subnetwork in Π; that is,

$$K' := \{k \in K : P^k \cap A^i \neq \phi \text{ and } P^k \cap A^j \neq \phi \text{ for each pair } A^i, A^j \in \Pi, i \neq j\}. \tag{3.28}$$

K' can be viewed as the set of 'problematic' commodities. If $k \in K'$ then different subnetworks might set different flow values for commodity k. However, if $k \notin K'$ then flow value for commodity k is unique and equals $\widehat{f}^k$.

The lower bounds are:

- z^1 :

$$z^1 := \sum_{k\notin K'} r^k \widehat{f}^k - \sum_{a\in A} \gamma_a(\widehat{x}_a). \tag{3.29}$$

  This corresponds to setting the flow of all commodities in K' to zero.

- z^2 :

$$z^2 := \sum_{k\in K} r^k \widehat{f}^k - \sum_{a\in A} \gamma_a(\widehat{x}_a). \tag{3.30}$$

  This corresponds to setting the flow of all commodities in K' to $\widehat{f}^k$.

- z^3 :

$$z^3 = \max_f \sum_{k \in K} r^k f^k - \sum_{a \in A} \gamma_a(\widehat{x}_a) \tag{3.31}$$

subject to

$$\sum_{k \in K : a \in P^k} f^k - \widehat{x}_a \leq 0, \qquad a \in A, \tag{3.32}$$

$$0 \leq f^k \leq \overline{b}^k, \qquad k \in K. \tag{3.33}$$

This bound is the optimal value of the LP obtained by solving WNDP with capacity decisions restricted to $\{\widehat{x}_a\}_{a \in A}$.

The following proposition holds:

Proposition 3.4

$$z^1 \leq z^2 \leq z^3 \leq z^*_{WNDP+} \leq z^*_D.$$

Proof.

1. $z^*_{WNDP+} \leq z^*_D$:

This follows directly from Proposition 3.2 and the definition of DUAL(Π).

2. $z^3 \leq z^*_{WNDP+}$:

The capacity levels $\{\widehat{x}_a\}_{a \in A}$ and the flow values $\{f^k\}_{k \in K}$ obtained by solving (3.31) - (3.33) constitute a feasible solution to WNDP+. z^3 is the value of this solution which must be less than or equal to the optimal value z^*_{WNDP+}.

3. $z^2 \leq z^3$:

This follows directly from the fact that the capacity levels $\{\widehat{x}_a\}_{a \in A}$ and the flow values $\{\widehat{f}^k\}_{k \in K}$ constitute a feasible solution to Problem (3.31) - (3.33).

4. $z^1 \leq z^2$:

This follows from the fact that $r^k \geq 0$ and $0 \leq \widehat{f}^k$ for all $k \in K'$. ■

Computing the tightest lower bound z^3 requires i) the availability of a feasible solution $\{\widehat{x}_a\}_{a\in A}$ and $\{\widehat{f}^{k,i}\}_{k\in K^i, i=1,\ldots,\pi}$ satisfying constraints (3.23) - (3.26), preferably the solution of WNDP+(II) for some λ, and ii) the solution of the linear program (3.31) - (3.33). The computation of z^2 is less intensive as it does not require solving the LP. Finally, z^1 has the advantage of yielding a bound on the optimality gap even before solving WNDP+(II), as stated in the following proposition:

Proposition 3.5

Consider any Lagrange multiplier vector λ satisfying properties (1) and (2) of Proposition 3.2. Then,

$$L(\lambda) - z^*_{WNDP+} \leq L(\lambda) - z^1 \leq \sum_{k\in K'} \overline{b}^k r^k.$$

Proof.

1. $L(\lambda) - z^*_{WNDP+} \leq L(\lambda) - z^1$:

This follows from Proposition 3.4 which states that $z^1 \leq z^*_{WNDP+}$.

2. $L(\lambda) - z^1 \leq \sum_{k\in K'} r^k \overline{b}^k$:

Properties (1) and (2) of Proposition 3.2 imply:

$$\begin{aligned} L(\lambda) &= \sum_{i=1}^{\pi} \sum_{k\in K^i} (\rho^{k,i} r^k - \lambda^{k,i})\, \widehat{f}^{k,i} - \sum_{i=1}^{\pi} \sum_{a\in A^i} \gamma_a(\widehat{x}_a) & (3.34) \\ &= \sum_{k\in K'} \sum_{i:k\in K^i} (\rho^{k,i} r^k - \lambda^{k,i})\, \widehat{f}^{k,i} + \sum_{k\notin K'} \sum_{i:k\in K^i} (\rho^{k,i} r^k - \lambda^{k,i})\, \widehat{f}^{k,i} - \sum_{i=1}^{\pi} \sum_{a\in A^i} \gamma_a(\widehat{x}_a) & (3.35) \\ &\leq \sum_{k\in K'} \sum_{i:k\in K^i} (\rho^{k,i} r^k - \lambda^{k,i})\, \overline{b}^k + \sum_{k\notin K'} \sum_{i:k\in K^i} (\rho^{k,i} r^k - \lambda^{k,i})\, \widehat{f}^k - \sum_{i=1}^{\pi} \sum_{a\in A^i} \gamma_a(\widehat{x}_a) & (3.36) \\ &= \sum_{k\in K'} \overline{b}^k r^k + \sum_{k\notin K'} \widehat{f}^k r^k - \sum_{i=1}^{\pi} \sum_{a\in A^i} \gamma_a(\widehat{x}_a). & (3.37) \end{aligned}$$

On the other hand,

$$z^1 = \sum_{k \notin K'} r^k \widehat{f}^k - \sum_{a \in A} \gamma_a(\widehat{x}_a). \tag{3.38}$$

The result follows by subtracting (3.38) from (3.37). ■

The term $r^k \overline{b}^k$ can be interpreted as the *value* of commodity k. The bound in Proposition 3.5 is therefore the total value of commodities whose paths span multiple subnetworks.

The Complete Partition Π^c, defined above, is the minimal partition with the property that $K' = \phi$. Therefore, a corollary of Proposition 3.5 is:

Corollary 3.1

Solving WNDP+(Π^c) is equivalent to solving WNDP+.

Proof.

For Π^c, $\sum_{k \in K'} \overline{b}^k r^k = 0$. Therefore, by Proposition 3.5, $L(\lambda) = z^*_{WNDP+}$ for any λ. The only λ that satisfies property (1) of Proposition 3.3 is the zero vector. In other words, solving WNDP+(Π^c) with any valid revenue allocation scheme solves WNDP+. ■

Intuitively, Π^c, is the finest partition of *flow-independent* subnetworks. The granularity of Π^c is a rough indication of the degree of coupling of flow decisions in the network. For weakly-coupled networks, Π^c consists of many small sized subnetworks rather than of a few large ones.

Definition 3.5 (Efficient partitions)

A partition Π is efficient if $\Pi \preceq \Pi^c$. It is inefficient otherwise.

When solving WNDP+(Π), we need only consider efficient partitions. The intuition behind this is that for any partition not nested within the complete partition, we can always construct a finer partition that provides the same degree of approximation. Finer partitions are more favorable computationally.

The next result confirms intuition that coarser partitions lead to smaller error bounds. Recall that $\Pi \preceq \Pi'$ means Π is nested within Π'.

Proposition 3.6

Consider two partition Π *and* Π' *defined by:*

$$\{A_1^1, ..., A_{p_1}^1, A_1^2, ...A_{p_2}^2, ..., A_1^\pi, ...A_{p_\pi}^\pi\} \equiv \Pi \preceq \Pi' \equiv \{A^1, A^2, ..., A^\pi\}$$

where $A_j^i \subseteq A^i$ *for all* $j = 1, ..., p_i$, $i = 1, ..., \pi$. *Let* $\lambda_j^{k,i}$ *be the Lagrange multiplier associated with commodity* k *and subnetwork* A_j^i *in Problem WNDP+(*Π*) and let* λ *denote the vector of all these Lagrange multipliers. Construct a vector* λ' *of Lagrange multipliers for Problem WNDP+(*Π'*) as follows:*

$$\lambda^{k,i} := \sum_{j=1}^{p_i} \lambda_j^{k,i} \text{ for all } k \in K \text{ and } i = 1, ..., \pi.$$

Then,

$$L(\lambda') \leq L(\lambda).$$

Proof.

We can assume, without loss of generality, that λ', and therefore λ, satisfies property (1) of Proposition 3.3 because otherwise $L(\lambda') = L(\lambda) = \infty$.

Given λ', let capacity values $\{\widehat{x}_a\}_{a\in A^i, i=1,...,\pi}$ and flow values $\{\widehat{f}^{k,i}\}_{k\in K^i, i=1,...,\pi}$ constitute an optimal solution to WNDP+(Π'). Construct a feasible solution to WNDP+(Π) as follows:

$$\begin{aligned} x_a &:= \widehat{x}_a, \quad \forall a \in A_j^i, \, j = 1, ..., p_i, \, i = 1, ..., \pi; \\ f_j^{k,i} &:= \widehat{f}^{k,i} \quad \forall k \in K_j^i, \, j = 1, ..., p_i, \, i = 1, ..., \pi, \end{aligned}$$

where K_j^i is the set of commodities whose paths include at least one arc in subnetwork A_j^i. It can be verified that the constructed solution is indeed feasible as it satisfies all constraints of Problem WNDP+(Π). The value of $L(\lambda)$ must be at least as large as the value of this feasible solution. Therefore,

$$\begin{aligned}
L(\lambda) &\geq \sum_{i=1}^{\pi}\sum_{j=1}^{p_i}\sum_{k\in K_j^i}(\rho_j^{k,i}r^k-\lambda_j^{k,i})\,f_j^{k,i}-\sum_{i=1}^{\pi}\sum_{j=1}^{p_i}\sum_{a\in A_j^i}\gamma_a(x_a)\\
&= \sum_{i=1}^{\pi}\sum_{j=1}^{p_i}\sum_{k\in K_j^i}(\rho_j^{k,i}r^k-\lambda_j^{k,i})\,\widehat{f}^{k,i}-\sum_{i=1}^{\pi}\sum_{j=1}^{p_i}\sum_{a\in A_j^i}\gamma_a(\widehat{x}_a)\\
&= \sum_{i=1}^{\pi}\sum_{k\in K^i}\sum_{j:k\in K_j^i}(\rho_j^{k,i}r^k-\lambda_j^{k,i})\,\widehat{f}^{k,i}-\sum_{i=1}^{\pi}\sum_{a\in A^i}\gamma_a(\widehat{x}_a)\\
&= \sum_{i=1}^{\pi}\sum_{k\in K^i}\widehat{f}^{k,i}\sum_{j:k\in K_j^i}(\rho_j^{k,i}r^k-\lambda_j^{k,i})-\sum_{i=1}^{\pi}\sum_{a\in A^i}\gamma_a(\widehat{x}_a)\\
&= \sum_{i=1}^{\pi}\sum_{k\in K^i}\widehat{f}^{k,i}(\rho^{k,i}r^k-\lambda^{k,i})-\sum_{i=1}^{\pi}\sum_{a\in A^i}\gamma_a(\widehat{x}_a)\\
&= L(\lambda').
\end{aligned}$$

■

3.5 Reformulation Step

The block angular structure of WNDP+(Π) and the observation that commodity flow variables induce excessive fractionality in the LP relaxation of WNDP+(Π) suggests that it might benefit from DW reformulation. In fact, the application of DW reformulation completely eliminates flow variables.

For reference, we re-state next the WNDP+(Π) formulation for any λ satisfying property

(1) of Proposition 3.3:

$$L(\lambda) = \max_{x,f} \sum_{i=1}^{\pi} \sum_{k \in K^i} (\rho^{k,i} r^k - \lambda^{k,i})\, f^{k,i} - \sum_{i=1}^{\pi} \sum_{a \in A^i} \gamma_a(x_a)$$

subject to

$$\sum_{i=1}^{\pi} D^i x^i \le d,$$

$$\sum_{k \in K^i : a \in P^k} f^{k,i} - x_a \le 0, \qquad a \in A^i,\ i = 1, ..., \pi,$$

$$0 \le f^{k,i} \le \bar{b}^k, \qquad k \in K^i,\ i = 1, ..., \pi,$$

$$x_a \in C_a, \qquad a \in A^i,\ i = 1, ..., \pi.$$

WNDP+(Π) has the simplifying property that the continuous variables do not appear in the coupling constraints. Therefore, in the terminology of Chapter 2, for each $x \in X$, we do not need to enumerate all extreme points E_x. Instead, we need only consider the extreme point with maximum objective function value. In the context of WNDP+(Π), this is equivalent to solving the best following local optimization problem for each subnetwork i and capacity configuration $\widehat{x} = \{\widehat{x}_a\}_{a \in A^i}$:

$$c^i(\widehat{x}) := \max_{f} \sum_{k \in K^i} (\rho^{k,i} r^k - \lambda^{k,i})\, f^{k,i} - \sum_{a \in A^i} \gamma_a(\widehat{x}_a) \tag{3.39}$$

subject to

$$\sum_{k \in K^i : a \in P^k} f^{k,i} - \widehat{x}_a \le 0, \qquad a \in A^i, \tag{3.40}$$

$$0 \le f^{k,i} \le \bar{b}^k, \qquad k \in K^i, \tag{3.41}$$

$$\widehat{x}_a \in C_a, \qquad a \in A^i.$$

Each of the preceding problems is an LP in the local flow variables and can be solved efficiently.

Let X^i denote the set of all capacity allocations to subnetwork i. If $A^i = \{a_1, a_2, ..., a_{m_i}\}$, then

$$X^i := \{(x_{a_1}, x_{a_2}, ..., x_{a_{m_i}}) : x_{a_j} \in C_{a_j} \text{ for all } j = 1, ..., m_i\}. \tag{3.42}$$

The DW reformulation of WNDP+(Π) is:

$$L(\lambda) = \max_{\mu} \sum_{i=1}^{\pi} \sum_{x \in X^i} c^i(x) \tag{3.43}$$

subject to

$$\sum_{i=1}^{\pi} \sum_{x \in X^i} (D^i x)\, \mu_x \leq d, \tag{3.44}$$

$$\sum_{x \in X^i} \mu_x = 1, \; i = 1, ..., \pi, \tag{3.45}$$

$$\mu_x \in \mathcal{B}, \quad \forall x \in X^i, \; i = 1, ..., \pi. \tag{3.46}$$

We denote this formulation by DW-WNDP+(Π). A number of properties are important to note:

- DW-WNDP+(Π) is a formulation in capacity variables only. Flow variables have been eliminated or, more precisely, incorporated in the objective function coefficients $c^i(x)$.
- The results of Chapter 2 show that the integer versions of DW-WNDP+(Π) and WNDP+(Π) are equivalent. However, the LP relaxation of DW-WNDP+(Π) is stronger than the LP relaxation of WNDP+(Π).
- The number of variables in DW-WNDP+(Π) is directly related to the granularity of partition Π. The coarser the partition, the greater the number of variables. The degree of approximation of DW-WNDP+(Π), equivalently WNDP+(Π), is inversely proportional to the coarseness of Π.

3.6 Solution Procedure

In this final section of this chapter we synthesize the ideas developed above into a proposed solution framework for WNDP+. The framework is outlined in Figure 3-3 and explained below.

Steps 1,2, and 3 in Figure 3-3 are initialization steps. The complete partition can be computed using Algorithm 3.1. A revenue allocation scheme is selected as described in

```
1. Compute Complete Partition, Πc
2. Set Revenue Allocation Scheme
3. Set i := 1
4. Select Efficient Partition Π = Πi where Πi ≤ Πc
    5. Select λ satisfying Proposition 3.3
        6. Solve DW-WNDP+(Π)
        7. Compute new lower bound on z*WNDP+
    Update λ
      until DUAL(Π) is 'sufficiently' solved
Update Π = Πi+1 where Πi ≤ Πi+1 ≤ Πc
  until DUAL(Π) is 'sufficiently' solved
```

Figure 3-3: Proposed iterative solution algorithm for solving WNDP+.

Section 3.4.2. Steps 6 and 7 form the core of the algorithm. In step 6, a DW reformulation of the approximate model WNDP+(Π) is solved. The bound in step 7 can be computed from any of the lower bounds z^1, z^2, or z^3 defined in (3.29), (3.30), or (3.31) - (3.33), respectively. It is worth noting that at any stage in the algorithm a feasible solution and an upper bound on the optimality gap are readily available.

The algorithm's progress can be visualized on a plot similar to that of Figure (3-4). The horizontal axis plots a sequence of successively nested partitions. It intersects the vertical axis at z^*_{WNDP+}. For a given efficient partition Π, the circles in the plot correspond to optimal values of Problem (3.22) - (3.26) for different choices of λ, and are therefore upper bounds on z^*_{WNDP+}. The squares in the plot are the lower bounds z^1, z^2, or z^3. The path of the algorithm alternates between vertical the arrows, representing iteration loop 5 in Figure 3-3, and the cross-partition arrows, representing iteration loop 4 in Figure 3-3.

Obtaining better approximations to WNDP+ can be performed in two complementary ways. The first is by solving DUAL(Π) to optimality or near optimality. The second is by using coarser efficient partitions. The ideal trade-off depends on various factors including number of variables in DW-WNDP+(Π), the availability of an efficient algorithm to solve the pricing problem, and the ability to solve the convex optimization problem DUAL(Π)

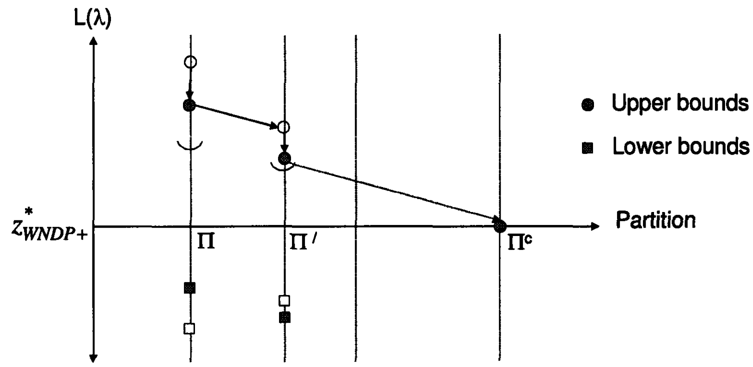

Figure 3-4: Progress path of solution algorithm

efficiently for λ. In Chapter 4, we report on a very simple implementation of the algorithm outlined here on an instance drawn from a major airline that yields results outperforming state-of-the-art models in the literature.

Chapter 4

Application: Airline Fleet Assignment with Enhanced Revenue Modeling

4.1 Introduction

4.1.1 The Fleet Assignment Problem

The *fleet assignment problem* addresses the question of how to best assign aircraft fleet types to an airline's schedule of flight legs. A flight leg is defined as a journey consisting of a single take off and a single landing, and thus constitutes the smallest unit of operation that can be assigned an aircraft. A flight schedule is a set of flight legs with specified departure and arrival times (arrival times can be fleet specific). A fleet assignment is a mapping from the set of flight legs to the set of fleet types. In this paper we assume that the flight schedule is fixed (see the next section for references to research that incorporate elements of schedule design into the fleet assignment problem).

The objective is to find a feasible fleet assignment that maximizes expected contribution as given by the simple relation:

$$\textit{expected contribution} = \textit{expected revenue - operating costs.} \tag{4.1}$$

Feasibility implies that each flight leg is assigned an aircraft, that this aircraft is available at the right airport and at the right time, that the number of aircraft employed of any particular fleet type does not exceed the number available, and that the fleet assignment meets any other operational constraints deemed important enough by the airline for consideration during the fleet assignment phase of the planning process.

Expected revenue is a function of passenger demand distribution, of aircraft seating capacity, and of the revenue management system employed by the airline. Passenger demand is itinerary and fare-class dependent. An itinerary is a sequence of one or more flight legs. Operating costs are fleet type and flight leg specific and include fuel, gate rental, and various other costs that are independent of the number of passengers travelling. Operating costs that depend on the number of passengers travelling are typically less significant in practice (but can be incorporated indirectly in our models through the expected revenue function).

The time horizon considered in this problem is usually 24 hours because most domestic airline operations in the US plan for a repeating daily schedule Monday through Friday.

Finding a feasible fleet assignment, let alone an optimal one, is a non-trivial task given the scale of operation of major airlines. Typical problem instances have more than 2000 flight legs flown by 10 fleet types serving at least 75,000 itineraries. A greater challenge lies in finding a good or near-optimal fleet assignment. Fleet types differ mainly in their seating capacities and their (leg-dependent) operating costs. Intuitively speaking, it is undesirable to assign a large aircraft to a flight leg with low passenger demand. However, given the nature of the problem, it may be advantageous in certain cases to have such assignments for reasons of connectivity or fleet availability. Moreover, because total passenger demand on a given flight leg can exceed the allocated aircraft capacity, it might be inevitable, or even desirable, to *spill* some passengers on certain itineraries. Spilling passengers means declining their demand or attempting to *recapture* them on alternative itineraries. Because passengers travelling on the same flight leg typically come from a mix of itineraries with different fares, the question of which passengers to spill and where to recapture them is in itself an optimization problem addressed by revenue management systems. Revenue management systems in common use employ a variety of rules governing the allocation of demand to capacity. These rules cannot always be conveniently captured in linear optimization models. For a survey of airline revenue management models, the reader is referred to McGill and van Ryzin [McR99].

4.1.2 Literature Review

The scale and complexity of fleet assignment problems coupled with their large cost implications have motivated the development of optimization-based methods to solve them. For an extensive overview of optimization models proposed for fleet assignment and other stages of the airline schedule planning process, the reader is referred to Gopalan and Talluri [GoT98]. The review below is representative, but by no means exhaustive.

Abara [Aba89] presents a formulation for a general flight network based on a partial enumeration of 'feasible turns', that is, connection opportunities between pairs of flight legs. The model was used by American Airlines in various studies to improve profitability.

Daskin and Panayotopoulos [DaP89] propose a 0-1 integer programming formulation for single hub networks. They employ a Lagrangian based procedure to obtain bounds on the objective function. The Lagrangian solution obtained is converted to a feasible fleet assignment and improved upon using heuristic methods.

Berge and Hopperstad [BeH93] address the fluctuating nature of passenger demand by presenting a procedure called *Demand Driven Dispatch* for dynamically reassigning fleet types to flight legs as better demand forecasts become available shortly before departure dates. They report 1-5% improvement in operating profits.

Hane, et al. [HBJ95] present the *Fleet Assignment Model (FAM)* which is a multicommodity network flow formulation with side constraints defined on an underlying 'time-line network'. The solution procedure described is based on preprocessing the network and applying the dual simplex method with steepest edge pricing coupled with specialized branching rules. The authors report fast solution times for realistically sized problems having 2500 flight legs and 11 fleet types. Some complexity results and other properties of FAM are presented by Gu, et al. [GJN94].

Farkas [Far96] draws attention to a limitation of FAM in accurately computing spill costs that could lead to sub-optimal solutions. Motivated by the need to properly handle network effects (see 4.2), Farkas [Far96] and Barnhart, Kniker, and Lohatepanont [BKL00] propose the *Itinerary-Based Fleet Assignment Model (IFAM)* which embeds a refinement of the spill optimization problem within FAM. Other research that incorporates passenger flow decisions into the fleet assignment process can be found in Jacobs, Johnson, and Smith [JJS99].

Rushmeier and Kontogiorgis [RuK97] present a formulation based on a 'connections-network' that allows modeling of various operational side constraints. They report annual savings of $15 million when implementing the model at USAir.

Success in solving fleet assignment problems has prompted the development of models integrating fleet assignment with other stages in the airline planning process such as schedule design, crew scheduling, and maintenance routing. Such models have been proposed by Barnhart, et al. [BBC98], Clarke, et al. [CHJ96], Cohn and Barnhart [CoB03], Cordeau, et al. [CSS01], Desaulniers, et al. [DDD97], Rexing, et al. [RBK00], and others.

4.1.3 Motivation

Whereas the basic fleet assignment models (e.g. of Hane, et al. and others) are currently solvable on large instances of practical fleet assignment problems, we share the concern expressed in the literature that the assumptions made on the revenue function are too simplistic to reflect the true complexity of current revenue management systems. Initial attempts to overcome these limitations (for example, the IFAM formulation) tend to come at the expense of poorer computational performance. It is not entirely clear, to the best of our knowledge, how even optimal solutions to our current tractable models actually perform in reality. Until this question is satisfactorily resolved, we need to i) strive to extend our models in creative ways to incorporate more realistic revenue functions, ii) devise practical solution procedures tailored to these models, and iii) derive, whenever possible, bounds on the optimality gap. This is especially critical when attempting to extend fleet assignment models to incorporate further elements of the airline planning process such as schedule design, crew-scheduling, and maintenance routing. Traditionally, these interrelated decision problems have been tackled separately or in a sequential manner. Current efforts to optimize these decisions simultaneously uncover the need for an accurate yet fast fleet assignment module. The reformulation approach and heuristic solution techniques developed in this section of the dissertation constitute an attempt in this direction. Specifically, the main contributions of this research are:

- *a strong linear mixed-integer formulation, called SFAM, for the airline fleet assignment problem that is able to incorporate more realistic revenue functions*; and
- *a flexible solution algorithm for SFAM that allows a trade-off between solution quality and solution time.* At any stage in the algorithm a feasible fleet assignment and an upper bound on the optimality gap are readily available.

The models and algorithms are first presented for general revenue functions highlighting any assumptions made when necessary. The specific case of the IFAM revenue function is then discussed where stronger results are derived.

This application provides a concrete example of the DW reformulation approach developed in Chapters 2 and 3. It also illustrates how the value of the application of the general technique is enhanced by tailoring it to the data and structural characteristics of a particular problem.

4.1.4 Outline

The remainder of this chapter is organized as follows. A general fleet assignment model is presented in section 4.2. The SFAM formulation is developed in section 4.3. Finally, we conclude in section 4.4 with limited computational results based on data drawn from a major U.S. airline.

4.2 Initial Formulation

4.2.1 Generic Model

Consider a problem instance where L denotes the set of all flight legs and F denotes the set of all fleet types. Let $x_{l,f}$ denote a binary decision variable defined as follows:

$$x_{l,f} = \begin{cases} 1 & \text{if fleet type } f \text{ is assigned to flight leg } l, \\ 0 & \text{otherwise.} \end{cases}$$

Let x_l denote the $|F|$-vector whose components are the decision variables corresponding to flight leg l and let x denote the $|L||F|$-vector corresponding to the concatenation of all x_l vectors. We call x the assignment vector. In addition, the problem may have other decision variables which we collectively refer to as auxiliary variables and denote by the vector y. Auxiliary variables may include maintenance, crew scheduling, or any other variables required for modeling purposes. The fleet assignment problem can be written generically as follows:

$$\begin{aligned} \text{maximize} \quad & \rho(x) - d\,x - h\,y \\ \text{subject to} \quad & A\,x + G\,y \leq b, \\ & x \in X = \{x \in B^{|L||F|} : e^t\,x_l = 1\ \forall l \in L\}, \\ & y \in Y, \end{aligned} \qquad \text{(GFAM)}$$

where b, d, h, A, and G are vectors and matrices of conformable dimensions, and e is an $|F|$-vector of 1s. Y is an arbitrary set defining feasible values of the auxiliary variables. The constraints $e\,x_l = 1\ \forall l \in L$ express the natural requirement that every flight leg needs to be assigned exactly one fleet type. Any other feasibility constraints expressable as linear inequalities can be incorporated into the formulation through an appropriate definition of auxiliary variables and model coefficients. When discussing the FAM model below, we give a specific example of what the constraints $A\,x + G\,y \leq b$ could represent. The objective function is a restatement of relation (4.1) where ρ gives the total expected revenue an airline achieves as a function of x and the term $d\,x + h\,y$ models the schedule's operational costs. Note the following assumption:

(A1) the revenue function, ρ, is a function of the assignment vector, x, only.

We believe that this assumption is justified for most practical purposes although it is possible that dependence on individual aircraft routing may exist (see, for example, Cordeau, et al. [CSS01] for a discussion of *through-revenues*). In parts of section 2.1, we impose further restrictions on the revenue function required for algorithmic purposes, and will highlight these assumptions. Apart from that, we make no further restrictions. In particular, ρ

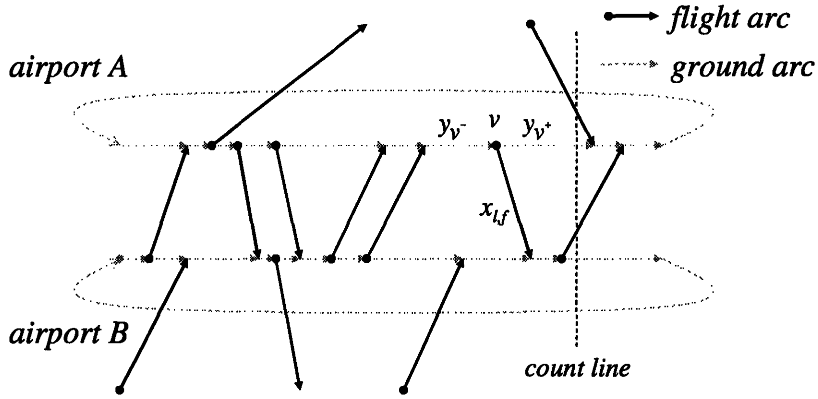

Figure 4-1: An example of a time-line network involving two airports.

does not have to be given explicitly. It could be an oracle that simulates the complexity of practical revenue management systems. We emphasize that the above formulation is highly flexible and accommodates most of the airline planning models proposed in the literature. We will refer to it as the *Generalized Fleet Assignment Model* (GFAM).

In the following two subsections we outline two concrete examples of fleet assignment models proposed in the literature that fall under the generic GFAM framework.

4.2.2 FAM

Since its introduction by Hane, et al. [HBJ95], the *Fleet Assignment Model* (FAM) and its variants have been employed extensively by major airlines to plan their fleet assignments with significant cost savings. We present below an overview of FAM and refer the reader to the original paper for details.

The underlying structure of FAM is a digraph called the time-line network (see figure 4-1 for an illustration).

Each node represents a particular airport at a particular time instance within the planning horizon (say, a departure or arrival time) for a particular fleet type. Denote the set of all nodes corresponding to fleet type f by N_f. Two types of directed arcs are used: flight arcs and ground arcs. A flight arc links two nodes of the same fleet type but different airport locations. Ground arcs (needed to model flow balance) link consecutive nodes for a particular airport and fleet type. Return ground arcs are required to link the last node of the day to the first node for every airport and fleet type, and thus ensure a repeating daily fleet assignment. For each node v let v^- denote the predecessor node defined by the tail of the ground arc entering v. Let I_v denote the set of flight legs inbound to node v and let O_v denote the set of flight legs outbound from node v. To be able to count the number of aircraft, an arbitrary common reference time is defined, called the count line. Let $T_F(f)$ denote the set of all flight legs that traverse the count line when assigned fleet type f. Similarly, let $T_N(f)$ denote the set of all nodes corresponding to fleet type f that have an exiting ground arc that traverses the count line. The total number of aircraft available of fleet type f is denoted by n_f.

The two types of arcs in the time-line network give rise to the two types of decision variables used in FAM: i) assignment variables, $x_{l,f}$, defined above and ii) ground arc auxiliary variables defined as follows:

y_v : the number of aircraft on the ground arc leaving node v.

Finally, let $r_{l,f}$ and $c_{l,f}$ denote, respectively, estimates of expected revenue and costs associated with assigning fleet type f to flight leg l. The FAM formulation is stated below:

$$
\begin{array}{lll}
\text{maximize} & \sum_{l\in L}\sum_{f\in F} r_{l,f}\, x_{l,f} - \sum_{l\in L}\sum_{f\in F} c_{l,f}\, x_{l,f} & \\
\text{subject to} & & \\
\text{(cover)} & \sum_{f\in F} x_{l,f} = 1, & \forall l \in L, \\
\text{(balance)} & y_{v^-} + \sum_{l\in I_v} x_{l,f} - y_v - \sum_{l\in O_v} x_{l,f} = 0, & \forall v \in N, \\
\text{(count)} & \sum_{v\in T_N(f)} y_v + \sum_{l\in T_F(f)} x_{l,f} \le n_f, & \forall f \in F, \\
\text{(integrality)} & x_{l,f} \in \{0,1\}, & \forall f \in F,\ \forall l \in L, \\
\text{(non-negativity)} & y_v \ge 0, & \forall v \in N_f,\ \forall f \in F.
\end{array}
\tag{FAM}
$$

The cover constraints ensure that each flight leg is assigned a fleet type. The balance constraints ensure conservation of aircraft flow by fleet type at each node. Count constraints ensure that for every fleet type the total number of aircraft in flight and on the ground at the count line does not exceed the total number of aircraft available of that fleet type. Integrality is explicitly imposed on the assignment variables while integrality of the ground arc variables implicitly follows from the formulation. Clearly, FAM is a special case of GFAM.

We note two main limitations of FAM:

1. the model does not incorporate any crew scheduling or maintenance considerations; and

2. the revenue function $\rho(x)$ is assumed to be linear in x. The coefficients $r_{l,f}$ can be estimated in various ways (see Kniker [Kni98] for an extensive discussion).

The linear assumption on revenue has been observed by various researchers to lead to inaccurate estimates of revenue. This is due to the fact that in multi-leg itineraries, capacity decisions on one flight leg affect the number of passengers spilled from that itinerary and, hence, from other flight legs. This phenomenon is called *network effects* and is described in detail in Farkas [Far96] and Kniker [Kni98]. A refined version of the revenue function is given next.

4.2.3 IFAM

The need to capture network effects has prompted the development of alternative fleet assignment models. Farkas [Far96] and Barnhart, et al. [BKL00] developed the Itinerary-Based Fleet Assignment Model (IFAM). IFAM uses decision variables that explicitly model the number of passengers booked on (or, alternatively, spilled from) each itinerary. Let P denote the set of all itineraries. For each $p \in P$, let D_p, v_p, and f_p denote respectively unconstrained (that is, maximum attainable) demand, number of booked passengers, and fare for itinerary p. Let $\delta_{p,l}$ be an indicator variable equal to 1 if itinerary p includes flight leg l and 0 otherwise. Finally, let CAP_f denote the capacity (in number of passengers) of fleet type f. The revenue function in IFAM is given by the following expression:

$$\begin{aligned} \rho(x) = \quad & \max \sum_{p \in P} f_p \, v_p & & \\ \text{subject to} \quad & \sum_{p \in P} \delta_{p,l} \, v_p \leq \sum_{f \in F} CAP_f \, x_{l,f}, & \forall l \in L, & \\ & 0 \leq v_p \leq D_p, & \forall p \in P. & \end{aligned} \tag{4.2}$$

Incorporating this expression into FAM instead of the linear term $\sum_{l \in L} \sum_{f \in F} r_{l,f} \, x_{l,f}$ yields the IFAM formulation which is a linear mixed-integer program.

IFAM solves the fleet assignment problem assuming an *idealized version* of revenue management. A version of IFAM that explicitly incorporates recapture decisions can be found in Barnhart, et al. [BKL00]. However, any improvements in the quality of the fleet assignment decisions achievable through this refinement in objective function come at an expense. IFAM's structure is similar to that of a network design problem, and hence, suffers from the notoriously weak bound provided by its associated LP relaxation. Relative to FAM, achieving optimal or near optimal solutions for the large-scale IFAM instances arising in the airline industry can be computationally time consuming. Moreover, our computational experience has shown that IFAM becomes intractable when extended to incorporate further elements in the airline schedule planning process.

IFAM is a special case of the WNDP+ formulation discussed in Chapter 3. We apply the ideas developed in that chapter to develop the SFAM formulation presented next. SFAM is an attempt to overcome the computational difficulties associated with IFAM, and to allow for modeling other revenue functions that might be better approximations of reality.

4.3 Extended Reformulation

In this section, we present the *Subnetwork Fleet Assignment Model* (SFAM) which is derived from GFAM through a two-step process:

1. an *approximation step* based on partitioning the set of flight legs into 'subnetworks'; and

2. a *reformulation step* based on a Dantzig-Wolfe decomposition.

The first step is equivalent to the WNDP+(II) formulation of Chapter 2. The second step is equivalent to the DW-WNDP+(II). One difference, however, is that the objective function of GFAM is generally non-linear in the capacity variables to reflect arbitrary revenue management models. For the sake of clarity, our discussion here is self- contained but draws upon results from Chapter 2 as needed.

4.3.1 Flight Leg / Itinerary Classification Schemes

For each flight leg $l \in L$, we assume the availability of an upper bound, d_l, on the number of passengers that can be booked. This can be viewed as the maximum attainable demand for the given flight leg regardless of capacity. For example, in IFAM this bound can be computed as $d_l = \sum_{p \in P} \delta_{p,l} D_p$. Let $\underline{CAP_l}$ and $\overline{CAP_l}$ denote, respectively, the smallest and largest aircraft capacity (measured in number of passengers) that can be assigned to flight leg l.

Definition 4.1 *Flight leg l is called unconstrained if $d_l \leq \underline{CAP_l}$. It is called potentially constrained if $d_l > \underline{CAP_l}$.*

Definition 4.2 *An itinerary is of Type r if it contains exactly r potentially constrained flight legs. The set of all itineraries of Type r will be denoted by P_r.*

4.3.2 Projections

Consider any vector $x \in \mathcal{B}^{|L||F|}$. Recall that x_l denotes the $|F|$-subvector whose components are the assignment variables corresponding to flight leg l. The projection of x on the subspace of variables defined by the flight legs in L_k is the subvector $x^k \in \mathcal{B}^{|L_k||F|}$ composed only of those elements of x corresponding to flight legs in L_k. Similarly, the projection of the set X on L_k is defined as $X^k := \{x^k : x \in X\}$.

4.3.3 Approximation Step

Revenue Function Approximation

We make the following important assumption on the revenue function ρ:

(A2) Given any partition $\Pi = \{L_1, ..., L_K\}$, we can construct a set of *local revenue functions* such that

$$\rho(x) \le \sum_{k=1}^{K} \rho_k(x^k) \quad \forall x \in X.$$

Assumption (A2) states that we can find an upper bound approximation of the revenue function that decouples by subnetwork. We first outline how this assumption may be satisfied in general and then show how it is satisfied for the special case of IFAM. Towards this end, we adopt the more specific notation $\rho(x, \eta)$ to refer to the revenue obtained given fleet assignment x *and* fare structure η. The fare structure is simply a vector in $\Re^{|P|}$ where the *pth* element is the fare of itinerary $p \in P$. Suppose we can find a set of *local fares* $\{\eta_k\}_{k=1}^{K}$such that

(A2a) $\rho(x, \eta) \le \sum_{k=1}^{K} \rho(x, \eta_k) \qquad \forall x \in X$; and

(A2b) the following optimization problem can be solved easily

$$\rho_k(x^k) := \max\{\rho(u, \eta_k) : u_l = x_l \; \forall l \in L_k \text{ and } \mathbf{1}^t x_l = 1 \; \forall l \in L \backslash L_k\}.$$

Then clearly, (A2a) and (A2b) imply (A2). The local revenue functions in (A2b) optimize revenues for a given subnetwork in a greedy fashion; that is, without regards to capacity limitations of other subnetworks. The details of how to perform the maximization in (A2b) depend on the particular revenue management system in use. To take a concrete example, consider the IFAM revenue function given in (4.2). Let the fare of itinerary p in local fare η_k be denoted $\eta_{k,p}$. Moreover, let $\delta_{p,L_k} = 1$ if itinerary p includes a flight leg in L_k, and 0 otherwise. Condition (A2a) is satisfied by (4.2) for any set of local fares where $\eta = \sum_{k=1}^{K} \eta_k$. In particular, select local fares satisfying (A2a) such that $\eta_{k,p} = 0$ whenever $\delta_{p,L_k} = 0$. The maximization in (A2b) is performed by solving the following LP (in the variables v_p):

$$\begin{aligned} \rho_k(x^k) = \quad & \max \sum_{p \in P} \eta_{k,p}\, v_p \\ \text{subject to} \quad & \sum_{p \in P} \delta_{p,l}\, v_p \le \sum_{f \in F} CAP_f\, x_{l,f}, \quad && \forall l \in L_k, \\ & \sum_{p \in P} \delta_{p,l}\, v_p \le \overline{CAP_l}, && \forall l \in L \backslash L_k, \\ & 0 \le v_p \le D_p, && \forall p \in P. \end{aligned} \tag{4.3}$$

GFAM(Π)

Given assumption (A2) on the revenue function, we can define the following approximation of GFAM which we denote by GFAM(Π). It is an intermediate step between GFAM and SFAM:

$$\begin{aligned} \text{maximize} \quad & \sum_{k=1}^{K} \rho_k(x^k) - \sum_{k=1}^{K} d^k\, x^k - h\, y \\ \text{subject to} \quad & \sum_{k=1}^{K} A^k\, x^k + G\, y \le b, \\ & x^k \in X^k, \quad k = 1, ..., K, \\ & y \in Y, \end{aligned} \tag{GFAM(Π)}$$

where d^k is the subvector of d corresponding to flight legs in L_k and A^k is the submatrix of A of the columns corresponding to the flight legs in L_k. Clearly, (x, y) is feasible for GFAM if and only if $(x^1, ..., x^K, y)$ is feasible for GFAM(Π). Hence, the approximation is in the objective function only.

Analysis of GFAM(Π)

In this section, we derive some properties of the quality of approximation of GFAM(Π) with respect to GFAM. Let $s = (x, y)$ denote any feasible solution to GFAM. Consider a flight leg partition Π^i. We distinguish between $z_i(s)$, the objective function value of GFAM(Π^i) corresponding to s, and $\tau(s)$, the objective function value of GFAM corresponding to s. We call $\tau(s)$ the 'true' contribution associated with s. The need for this distinction follows from the approximation to the true contribution inherent in GFAM(Π^i). Clearly, we seek a fleet assignment that maximizes $\tau(s)$, and not necessarily $z_i(s)$. Let s_i^* denote the fleet assignment obtained when solving GFAM(Π^i) and let $z_i^* := z_i(s_i^*)$ denote the corresponding optimal objective function value. Let τ^* denote the optimal contribution obtained by solving GFAM, that is, τ^* is the best true contribution achievable. Under Assumption (A2), the following relation is immediate:

$$z_i^* \geq \tau^* \geq \tau(s_i^*). \tag{4.4}$$

Clearly, GFAM(Π^F), with $\rho_1(.) := \rho(.)$ is identical to GFAM and, therefore, $z_F^* = \tau^*$. We now address the question of how the degree of approximation of GFAM(Π) depends on the choice of partition Π.

Proposition 4.1 *Consider $\Pi^i \preceq \Pi^j$ where $\Pi^i = \{L_1^i, ..., L_{K^i}^i\}$ and $\Pi^j = \{L_1^j, ..., L_{K^j}^j\}$. Let $\{\eta_k^i\}_{k=1}^{K^i}$ and $\{\eta_k^j\}_{k=1}^{K^j}$ be the set of local fares associated with Π^i and Π^j, respectively. Assume that the local fares have been selected such that $\rho(x, \eta_k^j) \leq \sum_{q:L_q^i \subseteq L_k^j} \rho(x, \eta_q^i)\ \forall x \in X$ (in other words, the revenue functions $\rho(x, \eta_k^j)$ satisfy Assumption (A2a)). Let the set of local revenue functions $\{\rho_k^i\}_{k=1}^{K^i}$ and $\{\rho_k^j\}_{k=1}^{K^j}$ corresponding to Π^i and Π^j, respectively, be computed according to (A2b). Then,*

$$z_i^* \geq z_j^*.$$

Proof. *From an optimal solution $(x^{j,1}, ..., x^{j,K^j}, y^j)$ to GFAM(Π^j) construct a solution $(x^{i,1}, ..., x^{i,K^i}, y^i)$ to GFAM(Π^i) as follows:*

$$\begin{aligned} x^{i,q} &: = \text{projection of } x^{j,k} \text{ on } L_q^i, \quad \forall L_q^i \subseteq L_k^j; \\ y^i &: = y^j. \end{aligned}$$

It is easy to verify that the constructed solution is feasible for GFAM(Π^i). The difference *in objective functions is:*

$$
\begin{aligned}
z_i^* - z_j^* &= \sum_{k=1}^{K^i} \rho_k^i(x^{i,k}) - \sum_{k=1}^{Kj} \rho_k^j(x^{j,k}) \\
&= \sum_{k=1}^{Kj} \{[\sum_{q:L_q^i \subseteq L_k^j} \rho_q^i(x^{i,q})] - \rho_k^j(x^{j,k})\} \\
&= \sum_{k=1}^{Kj} \{[\sum_{q:L_q^i \subseteq L_k^j} \rho(u_{k,i}, f_k^i)] - \rho(u_{k,j}, f_k^j)\} \\
&\geq \sum_{k=1}^{Kj} \{[\sum_{q:L_q^i \subseteq L_k^j} \rho(u_{k,j}, f_k^i)] - \rho(u_{k,j}, f_k^j)\} \\
&\geq 0
\end{aligned}
$$

where $u_{k,i}$ (respectively, $u_{k,j}$) is the maximizer of $\rho_k^i(x^{i,k})$ (respectively, $\rho_k^j(x^{j,k})$) in (A2b). ■

Recall the definition of complete partition, Π^c, introduced in Chapter 3. In the context of this problem, Π^c represents the minimal decomposition of flight legs into *spill-independent* subnetworks. Alternatively, Π^c is the *minimal* partition (with respect to nesting) where $z_c^* = \tau^*$.

The Special Case of IFAM

The results of the above section pertain to any revenue management system satisfying Assumptions (A1), (A2a), and (A2b). In the case of the IFAM revenue function given by (4.2) we can state stronger results. The next result provides a simple way of finding a bound on the optimality gap of GFAM(Π^i) without having to do any optimization. Let $\widehat{P}(\Pi^i)$ denote the set of itineraries of Type 2 or greater that contain potentially constrained flight legs belonging to more than one subnetwork under partition Π^i.

Proposition 4.2 *In the special case where GFAM is IFAM, and GFAM(Π^i) is defined by the local revenue functions given by (4.3),*

$$\tau^* \geq z_i^* - \sum_{p \in \widehat{P}(\Pi^i)} \eta_p \, D_p.$$

Proof. *This is a special case of Proposition 3.5.* ■

Note that $\widehat{P}(\Pi^c) = \phi$ by definition.

4.3.4 Reformulation Step

Full Enumeration Approach

Recall that X^k is the projection of the set X on the space of variables defined by subnetwork L_k. Let M^k denote the matrix whose columns are the elements of X^k. Note that if $|L_k| = 1$ then M^k is an identity matrix. We associate with each subnetwork L_k a decision vector $\omega^k \in B^{|X_k|}$. Hence, ω^k is a vector of *composite* decision variables. Each component of ω^k represents the assignment of fleet types to a *subnetwork* of flight legs. The vectors ω^k along with the vector of auxiliary variables y form the entire variable set of SFAM.

We now state the SFAM formulation:

$$\begin{aligned} \text{maximize} \quad & \sum_{k=1}^{K} \sum_{x^k \in X^k} \rho_k(x^k)\, \omega^k - \sum_{k=1}^{K} (d^k M^k)\, \omega^k - h\, y \\ \text{subject to} \quad & \sum_{k=1}^{K} (A^k M^k)\, \omega^k + G\, y \le b, \\ & e^k \omega^k = 1, \; k = 1, ..., K, \\ & \omega^k \in B^{|X^k|}, \; k = 1, ..., K, \\ & y \in Y, \end{aligned} \qquad \text{(SFAM}(\Pi))$$

where e^k is a $|W^k|$-row vector of 1s. To highlight the dependence of the formulation on the choice of partition, we use the notation SFAM(Π) to refer to SFAM applied to partition Π.

It follows from the results of Chapter 2 that the integer versions of SFAM(Π) and GFAM(Π) are equivalent. Note, however, that SFAM is a linear mixed-integer program regardless of the form of the revenue function. It should also be noted that even if the revenue function can be expressed linearly (perhaps, by adding extra variables / constraints to GFAM) then in general, SFAM will have a tighter continuous relaxation. This can considerably improve the model's tractability in LP-based branch-and-bound schemes.

Parsimonious Enumeration Approach

For a subnetwork consisting of $|L_k|$ flight legs, the full enumeration approach outlined above requires $|L_k||F|$ assignment decision variables. However, it is often possible to employ a smaller variables using *parsimonious variable enumeration* described in this subsection. To motivate the idea, consider the following example:

Example 4.1 *Subnetwork L_k consists of three flight legs $\{l_1, l_2, l_3\}$ that can be assigned any one of three fleet types $\{f_1, f_2, f_3\}$. The full enumeration approach requires $3^3 = 27$ variables regardless of the capacities of the fleet types. Suppose that these flights are constrained only under f_1. The parsimonious approach requires the following 13 variables only:*

θ	*Fleet Assignment*	θ	*Fleet Assignment*(l_1, l_2, l_3)
1	$(f_1, f_1, f_1) \to (l_1, l_2, l_3)$	8	$(f_2) \to (l_1)$
2	$(f_1, f_1) \to (l_1, l_2)$	9	$(f_2) \to (l_2)$
3	$(f_1, f_1) \to (l_1, l_3)$	10	$(f_2) \to (l_3)$
4	$(f_1, f_1) \to (l_2, l_3)$	11	$(f_3) \to (l_1)$
5	$(f_1) \to (l_1)$	12	$(f_3) \to (l_2)$
6	$(f_1) \to (l_2)$	13	$(f_3) \to (l_3)$
7	$(f_1) \to (l_3)$		

Variables $1-7$ correspond to all (partial) fleet assignments where all flight legs are constrained. Variables $8-13$ correspond to all elementary assignments where fleet type capacity exceeds the unconstrained demand of the flight leg. If $f_k \in \Re^{|P|}$ is the local fare structure associated with this subnetwork, then the coefficients of the above assignments are computed as specified by (A2b) with the provision that any flight leg that does not feature in an assignment is taken as uncapacitated. This guarantees that assignments not represented explicitly by the parsimonious variables are 'assembled' properly by the model from the above 13 variables. For example, the assignment $a' : (f_1, f_1, f_2) \to (l_1, l_2, l_3)$ will be represented by the model selecting the parsimonious assignments $a2 : (f_1, f_1) \to (l_1, l_2)$ and $a10 : (f_2) \to (l_3)$. The reader can check that the sum of the costs of $a1$ and $a10$ adds up to that of a'. It can also be checked that this dominates the cost of any other assembly such as $a5 : (f_1) \to (l_1)$,

$a6 : (f_1) \rightarrow (l_2),$ *and* $a10 : (f_2) \rightarrow (l_3).$

We generalize the above example as follows. We call a flight leg l *constrained under fleet type* f if $d_l > CAP_f$. The full set of parsimonious decision variables can be generated as follows. First, include all simple assignment variables. Second, include all other variables corresponding to (partial) fleet assignments where all flight legs are constrained (see the above example). The model is then free to 'patch' up those partial assignments to obtain the original one without any loss of accuracy. This apparently seems like introducing a greater number of variables. In fact, it almost always does the exact opposite because the partial assignments would serve as building blocks for other decomposed variables, thus slowing down the combinatorial explosion inherent in the full enumeration approach. We show next that this factorization of composite variables into elementary ones often leads to huge savings in the number of variables employed in medium to large sized subnetworks.

Let the number of assignment variables required by the full enumeration approach and the parsimonious approach be denoted N_1 and N_2 respectively. Let a_l denote the number of fleet types that can be assigned to flight leg l. Assume that the unconstrained demand for each flight leg $l \in L$ exceeds the capacity of exactly c_l fleet types ($c_l \leq a_l$). It is easy to verify that:

$$\begin{aligned} N_1 &= \prod_{l \in L} a_l; \text{ and} \\ N_2 &= \sum_{l \in L}(a_l - c_l) + \left[\prod_{l \in L}(c_l + 1)\right] - 1. \end{aligned} \tag{4.5}$$

To gain insight into the savings in the number of assignment variables, consider the case where $a_l = a$, and $c_l = pa \quad \forall l \in L,\ p \in [0,1]$. Denote the number of flight legs by n. In this case,

$$\frac{N_2}{N_1} = \frac{na(1-p) - 1}{a^n} + (\frac{1}{a} + p)^n.$$

Table 4.1 gives representative values of this ratio.

n	a	p		
		0	0.5	1
3	2	0.75	1.25	3.25
	5	0.12	0.40	1.72
	10	0.03	0.23	1.33
5	2	0.31	1.125	7.56
	5	< 0.01	0.17	2.49
	10	< 0.01	0.08	1.61
15	2	< 0.001	1.00	437.89
	5	< 0.001	< 0.01	15.41
	10	< 0.001	< 0.001	4.18

Table 4.1: Representative savings due to the parsimonious approach.

Number of fleet types	**9**
Number of flight legs	**2044**
Percentage of potentially constrained flight legs	17%
Number of itineraries	**76741**
Percentage of Type 0 itineraries	69%
Percentage of Type 1 itineraries	29%
Percentage of Type 2 or greater itineraries	2%

Table 4.2: SFAM test instance: summary statistics.

It is clear that for flight legs constrained by a relatively small number of fleet types, the savings in variables as a result of employing the parsimonious approach is very substantial. This is especially true as the number of flight legs and / or fleet types increases. Whichever approach is used, the results and solution algorithms of SFAM presented in this Chapter are unchanged.

4.4 Proof-of-Concept Results

In this section we present results obtained by applying SFAM to a data set drawn from a major U.S. airline where the IFAM revenue function is assumed to hold. Summary statistics of the data set are given in table 4.2.

The complete partition for this data set has a total of 8 subnetworks of size larger than 1. The largest of these subnetworks has 275 flight legs, the rest have a small number of flight legs each.

Given the presence of a prohibitively large subnetwork in Π^c, it is necessary to employ

	FAM	IFAM	SFAM
Problem size			
Number of columns	18,487	77,284	96,148
Number of rows	7,827	10,905	7,847
Number of non-zero entries	50,034	128,864	81,920
Solution time [sec]			
Time in branch and bound	877	>2113	365
Total time	974	>2296	771
Difference between LP relaxation and best IP solution objective function values	36	235,649	891
Contribution [$/day]	21,178,815	21,066,811	21,227,196
Improvement over FAM [$/year]	0	(29,233,044)	12,627,441

Table 4.3: A comparison between FAM, IFAM and SFAM results.

finer partitions. A solution procedure that is a simplified version of that outlined in Figure 3-3 of Chapter 3 was used. In particular, no iteration in steps 4 and 5 took place. The Lagrange multiplier vector λ of step 5 was always set to 0. The efficient partition at the beginning of step 4 was selected as follows: Starting with the complete partition, lowest fares itineraries were incrementally removed from those subnetworks that had many flight legs. The outcome was a breakdown of the complete partition into one with 21 subnetworks of size larger than 1. The largest of these subnetworks had 6 flight legs. Hence the complete partition was fragmented into a finer partition to decrease the number of composite columns that need to be generated. A *single iteration* of SFAM was then solved using CPLEX 6.5 on a HPC 3000 machine. The results obtained are reported in table 4.3 and compared to the results obtained from implementing FAM and IFAM on the same problem instance.[1]

The total solution time reported for SFAM includes the pre-processing time required for computing the complete partition, constructing a finer partition, and calculating the spill cost objective function coefficients. This pre-processing time was no more than 300 seconds.

The number of composite columns added was 77,661. The generation of these columns was performed employing the parsimonious approach. Using this approach, the number of composite variables that needs to be generated is only a small fraction of the number

[1]The computational experiments reported in this section were performed by Manoj Lohatepanont. The reader is referred to [Loh01] for a more extensive set of results.

required to enumerate all possible fleet assignments for the specified partition. In fact, the parsimonious approach in this instance resulted in a 90% saving of columns required.

The tightness of the LP relaxation of SFAM is clearly demonstrated by this problem instance. The integrality gap of the SFAM formulation was 0.3% of that displayed by IFAM.

The solution time reported for IFAM was the time taken to generate the best integer solution reported. CPLEX was left to run for several hours afterwards with no improvement in objective function value. IFAM failed, therefore, to generate an optimal solution within an acceptable solution time. In fact, the best integer solution generated had a lower contribution value than the one obtained by FAM.

The difference between the optimal objective function value of SFAM and the true contribution associated with the solution obtained provides a bound on the optimality gap of SFAM. This difference was $94,155 per day or equivalently an optimality gap of 0.4%.

SFAM has, therefore, successfully solved a large instance of the fleet assignment problem in a reasonable amount of time while incorporating networks effects. The result is a fleet assignment better than that obtained by FAM and IFAM. While not exhaustive, this computational experiment is indicative of SFAM's tractability and applicability to practical size fleet assignment and extended fleet assignment problems. It also demonstrates that a very simplified version of the algorithm developed in Chapter 3 can yield significant result. More sophisticated implementations can yield more savings.

Chapter 5

Part I Summary and Future Directions

5.1 Summary of Contributions

In our view, the main contributions of this research are:

1. An extensive analysis of the complexity and structure of a weakly-coupled network design problem, WNDP+;

2. A flexible solution framework for WNDP+ that provides a trade-off between solution quality and solution time;

3. The application of this framework to develop a novel formulation for the airline fleet assignment problem that is able to incorporate more realistic revenue models.

5.2 Future Directions

This research can be developed further along a number of possible directions:

- How can the results of Chapter 3 be extended to network design problems characterized by a 'small' number of paths for each commodity?

- What applications other than capacity provision in revenue management systems can be cast as WNDP+?
- We need to perform a more extensive empirical assessment of the performance of the algorithms proposed.

Bibliography

[Aba89] J. ABARA. 1989. "Applying Integer Linear Programming to the Fleet Assignment Problem". *Interfaces,* **19** (4), 20-28.

[App69] L. H. APPELGREN. 1969. "A Column Generation Algorithm for a Ship Scheduling Problem". *Transportation Science*, **3**, 53-68.

[App71] L. H. APPELGREN. 1971. "Integer Programming Methods for a Vessel Scheduling Problem". *Transportation Science*, **35**, 64-78.

[ABW02] A. P. ARMACOST, C BARNHART, and K. A. WARE. 2002. "Composite Variable Formulations for Express Shipment Service Network Design". *Transportation Science*, **36**, 1-20.

[BMM97] A. BALAKRISHNAN, T. L. MAGNANTI, and P. MIRCHANDANI. 1997. "Network Design", Chapter 18 in *Annotated Bibliographies in Combinatorial Optimization*, edited by Dell'Amico, Maffioli, and Martello.

[BMM95a] M. O. BALL, T. L. MAGNANTI, C. L. MONMA, and G. L. NEMHAUSER, Eds. 1995. *Handbooks in Operations Research and Management Science, Vol 7: Network Models.* North-Holland.

[BMM95b] M. O. BALL, T. L. MAGNANTI, C. L. MONMA, and G. L. NEMHAUSER, Eds. 1995. *Handbooks in Operations Research and Management Science, Vol 8: Network Routing.* North-Holland.

[BBC98] C. BARNHART, N. BOLAND, L. CLARKE, E. JOHNSON, G. NEMHAUSER, and R. SHENOI. 1998. "Flight String Models for Aircraft Fleeting and Routing". *Transportation Science,* **32** (3), 208-220.

[BHJ95] C. BARNHART, C. A. HANE, E. L. JOHNSON, and G. SIGISMONDI. 1995. "A Column Generation and Partitioning Approach for Multicommodity Flow Problems". *Telecommunication Systems*, **3**, 239-258.

[BJN98] C. BARNHART, E. L. JOHNSON, G. L. NEMHAUSER, M. W. P. SAVELSBERGH, and P. H. VANCE. 1998. "Branch-and-Price: Column Generation for Solving Huge Integer Programs". *Operations Research*, **46**, 316-329.

[BHV00] C. BARNHART, C. A. HANE, and P. H. VANCE. 2000. "Using Branch-and-Price-and-Cut to Solve Origin-Destination Integer Multicommodity Flow Problems". *Operations Research*, **48**, 318-326.

[BKL00] C. BARNHART, T. KNIKER, M. LOHATEPANONT. 2000. "Itinerary-Based Airline Fleet Assignment". *Transportation Science*, **36** (2), 199-217.

[BKK02] C. BARNHART, N. KRISHNAN, D. KIM, and K. WARE. 2002. "Network Design for Express Shipment Delivery". *Computational Optimization and Applications*, **21**, 239-262.

[BeH93] M. BERGE and C. HOPPERSTAD. 1993. "Demand Driven Dispatch: A Method for Dynamic Aircraft Capacity Asignment, Models and Algorithms". *Operations Research,* **41** (1), 153-168.

[BeT97] D. BERTSIMAS and J. N. TSITSIKLIS. 1997. *Introduction to Linear Optimization.* Athena Scientific.

[CHJ96] L. CLARKE, C. HANE, E. JOHNSON, and G. NEMHAUSER. 1996. "Maintenance and Crew Considerations in Fleet Assignment". *Transportation Science,* **30** (3), 249-260.

[CoB03] A. M. COHN and C. BARNHART. 2003. "Improving Crew Scheduling By Incorporating Key Maintenance Routing Decisions". *Operations Research*, **51** (3), 387-396.

[CSS01] J. F. CORDEAU, G. STOJKOVIC, F. SOUMIS, and J. DESROSIERS. 2001. "Benders Decomposition for Simultaneous Aircraft Routing and Crew Scheduling". *Transportation Science,* **35** (4), 375-388.

[CrR87] T. G. CRAINIC and J. M. ROUSSEAU. 1987. "The Column Generation Principle and the Airline Crew Scheduling Problem". *INFOR*, **25** (2), 136-151.

[DaW60] G. B. DANTZIG and P. WOLFE. 1960. "Decomposition Principle for Linear Programs". *Operations Research*, **8**, 101-111.

[DaP89] M. DASKIN and N. PANAYOTOPOULOS. 1989. "A Lagrangian Relaxation Approach to Assigning Aircraft Routes in Hub and Spoke Networks". *Transportation Science,* **23** (2), 91-99.

[DDD97] G. DESAULNIERS, J. DESROSIERS, Y. DUMAS, M. M. SOLOMON, and F. SOUMIS. 1997. "Daily Aircraft Routing and Scheduling", *Management Science,* **43**, 841-855

[DeS89] M. DESROCHERS and F. SOUMIS. 1989. "A Column Generation Approach to the Urban Transit Crew Scheduling Problem". *Transportation Science*, **23**, 1-13.

[DDS92] J. DESROCHERS, J. DESROSIERS, and M. SOLOMON. 1992. "A New Optimization Algorithm for the Vehicle-Routing Problem with Time Windows". *Operations Research*, **40**, 342-354.

[DSD84] J. DESROSIERS, F. SOUMIS, and M. DESROCHERS. 1984. "Routing with Time Windows by Column Generation". *Networks*, **14**, 545-565.

[DDS95] J. DESROSIERS, Y. DUMAS, M. M. SOLOMON, and F. SOUMIS. 1995. "Time Constrained Routing and Scheduling", Chapter 2 in *Handbooks of Operations Research and Management Science, Volume* **8**, edited by Ball et al.

[DDS91] Y. DUMAS, J. DESROSIERS, and F. SOUMIS. 1991. "The Pickup and Delivery Problem with Time Windows". *European Journal of Operational Research*, **54**, 7-22.

[Far96] A. FARKAS. 1996. *The Influence of Network Effects and Yield Management on Airline Fleet Assignment Decisions.* Ph.D. Disseration, Massachusetts Institute of Technology.

[FoF58] L. R. FORD and D. R. FULKERSON. 1958. "A Suggested Computation for Maximal Multi-commodity Network Flows". *Management Science*, **5**, 97-101.

[GSM99] M. GAMACHE, F. SOUMIS, G. MARQUIS, and J. DESROSIERS. 1999. "A Column Generation Approach for Large-Scale Aircrew Rostering Problem". *Operations Research*, **47**, 247-263.

[GaJ79] M. R. GAREY and D. S. JOHNSON. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman.

[Geo70a] A. M. GEOFFRION. 1970. "Elements of Large-Scale Mathematical Programming: Part I: Concepts". *Management Science*, **16** (11), 652-675.

[Geo70b] A. M. GEOFFRION. 1970. "Elements of Large-Scale Mathematical Programming: Part II: Synthesis of Algorithms and Bibliography". *Management Science*, **16** (11), 676-691.

[GiG61] P. C. GILMORE and R. E. GOMORY. 1961. "A Linear Programming Approach to the Cutting Stock Problem". *Operations Research*, **9**, 849-859.

[GiG63] P. C. GILMORE and R. E. GOMORY. 1963. "A Linear Programming Approach to the Cutting Stock Problem - Part II". *Operations Research*, **11**, 863-888.

[GoT98] R. GOPALAN and K. T. TALLURI. 1998. "Mathematical Models in Airline Schedule Planning: A Survey". *Annals of Operations Research,* **76**, 155-185.

[GJN94] Z. GU, E. JOHNSON, G. NEMHAUSER, and Y. WANG. 1994. "Some Properties of the Airline Fleet Assignment Problem". *Operations Research Letters,* **15**, 59-71.

[HBJ95] C. A. HANE, C. BARNHART, E. L. JOHNSON, R. E. MARSTEN, G. L. NEMHAUSER, and G. SIGISMONDI. 1995. "The Fleet Assignment Problem: Solving a Large-Scale Integer Program". *Mathematical Programming,* **70**, 211-232.

[Ho87] J. K. HO. 1987. "Recent Advances in the Decomposition Approach to Linear Programming". *Mathematical Programming Study*, **31**, 119-127.

[JJS99] T. L. JACOBS, E. L. JOHNSON, and B. C. SMITH. 1999. "O&D FAM: Incorporating Passenger Flows into the Fleeting Process". AGIFORS Symposium, New Orleans, LA.

[JLF93] K. L. JONES, I. J. LUSTIG, J. M. FARVOLDEN, and W. B. POWELL. 1993. "Multicommodity Network Flows - The Impact of Formulation on Decomposition". *Mathematical Programming*, **62**, 95-117.

[Kni98] T. S. KNIKER. 1998. *Itinerary-Based Airline Fleet Assignment.* Ph. D. Dissertation, Massachusetts Institute of Technology.

[Las70] L. S. LASDON. 1970. *Optimization Theory for Large Systems.* Macmillan.

[LMO89] S. LAVOIE, M. MINOUX, and E. ODIER. 1989. "A New Approach for Crew Pairing Problems by Column Generation with an Application to Air Transportation". *European Journal of Operational Research*, **35**, 45-58.

[Loh01] M. LOHATEPANONT. 2001. *Fleet Assignment and Schedule Design: Integrated Models and Algorithms.* Ph.D. Dissertation, Massachusetts Institute of Technology.

[McR99] J. I. MCGILL. and G. J. VAN RYZIN. 1999. "Revenue Management: Research Overview and Prospects". *Transportation Science,* **33**, 233-256.

[MSW76] T. L. MAGNANTI, J. F. SHAPIRO, and M. H. WAGNER. 1976. "Generalized Linear Programming Solves the Dual". *Management Science*, **22**, 1195-1203.

[MaW84] T. L. MAGNANTI and R. T. WONG. 1984. "Network Design and Transportation Planning - Models and Algorithms". *Transportation Science*, **18**, 1-55.

[Mar87] R. K. MARTIN. 1987. "Generating Alternative Mixed-Integer Programming Models Using Variable Redefinition". *Operations Research*, **35**, 820-831.

[Mar99] R. K. MARTIN. 1999. *Large-Scale Linear and Integer Optimization: A Unified Approach.* Kluwer Academic Publishers.

[Min89] M. MINOUX. 1989. "Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications". *Networks*, **19**, 313-360.

[RBK00] B. REXING, C. BARNHART, T. KNIKER, A. JARRAH, and N. KRISHNAMURTHY. 2000. "Airline Fleet Assignment with Time Windows". *Transportation Science,* **34** (1), 1-20.

[RuK97] R. RUSHMEIER and S. KONTOGIORGIS. 1997. "Advances in the Optimization of Airline Fleet Assignment". *Transportation Science,* **31** (2), 159-169.

[Sch03] A. SCHRIJVER. 2003. *Combinatorial Optimization: Polyhedra and Efficiency.* Springer.

[ShA90] H. D. SHERLAI and W. P. ADAMS. 1990. "A Hierarchy of Relaxations Between the Continuous and Convex-Hull Representations for Zero-One Programming-Problems". *SIAM Journal of Discrete Mathematics*, **3**, 411-430.

[ShA94] H. D. SHERLAI and W. P. ADAMS. 1994. "A Hierarchy of Relaxations and Convex Hull Characterizations for Mixed-Integer Zero-One Programming Problems". *Discrete Applied Mathematics*, **52**, 83-106.

[Tod90] M. J. TODD. 1990. "A Dantzig-Wolfe-Like Variant of Karmarkar's Interior-Point Linear-Programming Algorithm". *Operations Research*, **38**, 1006-1018.

[VaW96] F. VANDERBECK and L. A. WOLSEY. 1996. "An Exact Algorithm for IP Column Generation". *Operations Research Letters*, **19**, 151-159.

[Van00] F. VANDERBECK. 2000. "On Dantzig-Wolfe Decomposition in Integer Programming and Ways to Perform Branching in a Branch-and-Price Algorithm". *Operations Research*, **48**, 111-128.

[Wol98] L. WOLSEY. 1998. *Integer Programming.* Wiley-Interscience.

Part II

Performance of Greedy Algorithms for Maximizing Submodular Functions over Independence Systems

Chapter 6

Introduction to Part II

A large class of optimization algorithms are based on the natural idea of constructing a good feasible solution by solving a sequence of lower-dimension optimization problems. The complex task of finding a global optimum is decomposed into a sequence of *myopic* local searches. Each local problem in the sequence might be solved optimally or solved approximately by finding a 'good' solution. Examples of algorithms falling within this class include the simplex method for linear programs, augmenting path algorithms for maximum flow problems, and Kruskal's algorithm for minimum spanning trees.

Although we will not attempt to provide a rigorous argument here, it might be useful to observe that some of these algorithms (though, apparently, not all) possess the additional property of being *augmentation* algorithms; that is, roughly speaking, they 'build' a solution in a sequence of augmentation steps without revisiting past decisions. Kruskal's algorithm clearly falls within this category. Augmenting path algorithms also fall within this category when viewed in the space of residual network paths. Clearly, the simplicity provided by myopic augmentation algorithms provides computational advantages, yet it narrows down the range of problems for which they are suitable.

Arguably, such algorithms are most likely to succeed for convex problems where it suffices to find a local optimum. However, even for this special case, convergence to a local optimum might not be guaranteed. An example is the application of a generic augmenting path algorithm to the max flow problem with irrational data. Even when they are guaranteed to converge, different implementations can differ significantly in running time complexity.

Now consider nonconvex problems (including discrete ones). It is not difficult to point out numerous cases where myopic augmentation algorithms either take exponential time to find optimal solutions or perform arbitrarily badly when limited to polynomial running time. The fact that they do sometimes perform to within a constant factor of optimality is the exception, not the rule.

In this part of the dissertation, we investigate the performance of a family of myopic augmentation algorithms, which we refer to as GENERALIZED GREEDY, when applied to the abstract discrete problem of maximizing a submodular function over an independence system. This problem encompasses many well-known combinatorial problems of theoretical as well as practical interest. GENERALIZED GREEDY incrementally augments the current solution by adding subsets of arbitrary, possibly variable, cardinality. On one extreme it reduces to the well known standard best-in greedy algorithm, which we refer to as STANDARD GREEDY. On the other extreme it is synonymous with complete enumeration. It spans a range of tractability-versus-optimality trade-offs. Building on pioneering work by Conforti, Cornuéjols, Fisher, Nemhauser, Wolsey and others, we analyze the worst-case performance of such an algorithm in terms of its parameters. We then define a continuous relaxation of the original problem and show that some of the derived bounds apply with respect to the relaxed problem. This extends, and in some cases strengthens, previously known results for the standard best-in greedy. The bounds help us identify problem instances where such an algorithm performs well.

In this chapter, we define the problem formally, review some preliminary results, present a selection of motivating applications, and state the STANDARD GREEDY algorithm. In Chapter 7, we survey relevant complexity and approximation results. In Chapter 8, we define GENERALIZED GREEDY and analyze its performance for matroids in Chapter 9. Chapter 10 extends some of the bounds to a continuous relaxation of the discrete problem and relaxes the matroid assumption to derive a new bound for general independence systems. Finally, we conclude in Chapter 11 with a summary of contributions and an exposition of open problems.

6.1 Problem Statement

The definitions presented in this section can be found in many standard expositions. For instance, the reader is referred to [Lov83], [NeW88], and [Sch03].

A set system $I := (N, \mathcal{F})$, where N is a finite set of cardinality n and $\mathcal{F}$ is a collection of subsets of N, is an independence system if it satisfies the following properties:

(P1) $\phi \in \mathcal{F}$;

(P2) $A \subseteq B \in \mathcal{F}$ implies $A \in \mathcal{F}$.

An independence system is a matroid if, in addition to (P1) and (P2), it satisfies the following property:

(P3) If $A,\ B \in \mathcal{F}$ and $|A| < |B|$ then there exists $e \in B \backslash A$ such that $A \cup \{e\} \in \mathcal{F}$.

An equivalent way of stating property (P3) is:

(P3$'$) For all $C \subseteq N$, if A and B are maximal in C with respect to $\mathcal{F}$ then $|A| = |B|$.

Properties (P2), (P3), and (P3$'$) are commonly referred to as the "hereditary", "growth", and "maximal = maximum" properties, respectively.

A real-valued set function $z : 2^N \rightarrow \mathcal{R}$ is submodular if and only if:

(P4) $z(A) + z(B) \geq z(A \cup B) + z(A \cap B)$ for all $A, B \subseteq N$.

Note that affine functions are submodular, with equality holding in (P4) for each pair of sets $A, B \subseteq N$.

For any $A \subseteq N$, define the *marginal value function* of z with respect to A by

$$\rho_A(B) := z(A \cup B) - z(A) \quad \forall B \subseteq N. \tag{6.1}$$

It can be shown (see Schrijver [Sch03]) that an equivalent definition of submodularity is given by:

(P4$'$) $\rho_B(\{j\}) \leq \rho_A(\{j\}) \quad \forall A \subseteq B \subseteq N$ and $j \in N \backslash B$.

Therefore, submodularity, can be viewed as a discrete analogue of concavity. It frequently arises in systems exhibiting "economies of scale"; that is, non-increasing marginal returns.

We refer to the function z as normalized when:

(P5) $z(\phi) = 0$.

We refer to it as nondecreasing when:

(P6) $A \subseteq B \subseteq N$ implies $z(A) \leq z(B)$.

The main problem addressed in this part of the thesis can be stated as follows:

$$z^* := \max\{z(S) : S \in \mathcal{F}\}. \tag{P}$$

where $z : 2^N \rightarrow \mathcal{R}$ is a normalized nondecreasing submodular function and $\mathcal{F}$ is an independence system (in certain sections we will restrict attention to matroids). We refer to z as the *objective function* and to $\mathcal{F}$ as the set of *feasible solutions.* A number of well-known combinatorial optimization problems can be cast as Problem (P). We review some of these problems and their applications in Section 6.3.

The following example is an instance of Problem (P) and illustrates that a strict local optimum is not necessarily a global optimum. A strict local optimum is interpreted as a solution that deteriorates through any single element exchange. This fact, in our view, is one of the reasons why it is difficult to design good algorithms that solve Problem (P).

Example 6.1

$N = \{e_1, e_2, e_3, e_4\}$. *The function z is defined in the table below.*

A	$z(A)$	A	$z(A)$
ϕ	0	$\{e_2, e_3\}$	1.4
$\{e_1\}$	1	$\{e_2, e_4\}$	1.5
$\{e_2\}$	1	$\{e_3, e_4\}$	1.4
$\{e_3\}$	1	$\{e_1, e_2, e_3\}$	1.8
$\{e_4\}$	1	$\{e_1, e_2, e_4\}$	1.8
$\{e_1, e_2\}$	1.4	$\{e_1, e_3, e_4\}$	1.8
$\{e_1, e_3\}$	1.7	$\{e_2, e_3, e_4\}$	1.8
$\{e_1, e_4\}$	1.4	N	1.8

It can be verified that this function is normalized, nondecreasing, and submodular. Now define the matroid $I := (N, \mathcal{F})$ by all subsets of N having cardinality 2 or less; that is

$$\mathcal{F} = \{\phi, \{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_1, e_2\}, \{e_1, e_3\}, \{e_1, e_4\}, \{e_2, e_3\}, \{e_2, e_4\}, \{e_3, e_4\}\}.$$

It can be verified that this indeed defines a matroid (in fact it is a uniform matroid in the terminology defined in the next section). Now consider the solution $\{e_2, e_4\}$ of value 1.5. This solution deteriorates through by any single element exchange with the set $\{e_1, e_3\}$. Therefore, it can be considered a strict local optimum. However, the unique global optimum is $\{e_1, e_3\}$ with value 1.7. It is also worth noting that the solution $\{e_2, e_4\}$ could be a solution produced by the STANDARD GREEDY *algorithm defined in Section 6.4. This shows that the combination of greedy and local exchange might perform no better than greedy alone (an observation that holds even in some worst-case instances). This property is also shared by the* GENERALIZED GREEDY *algorithm defined in Chapter 8.*

We conclude this section by noting that the normalization property (P5) does not impose any loss of generality. Given any nondecreasing submodular function z' with $z'(\phi) = c$, we can construct a normalized nondecreasing submodular function $z(S) := z'(S) - c$ for all $S \subseteq N$. In this case, the performance guarantees derived should be interpreted as bounds on the ratio $(z^* - c)/(z^G - c)$.

6.2 Preliminaries

In this section we establish various elementary facts and characterizations of submodular functions and independence systems.

The following proposition is easy to establish from the definition of the marginal objective function.

Proposition 6.1

If $z : 2^N \to \mathcal{R}$ is nondecreasing submodular then, for any $A \subseteq N$, the marginal value function $\rho_A(\cdot)$ defined by (6.1) is normalized, nondecreasing, and submodular.

The next proposition will be useful in establishing the bounds derived in Section 9.

Proposition 6.2

For any normalized submodular function $z : 2^N \to \mathcal{R}$, any nonempty subset A of N and any integer $d \in \{1, ..., |A|\}$, if $\max\{z(B) : B \subseteq A, |B| = d\} \leq b$, then

$$z(A) \leq \frac{|A|}{d} b.$$

Proof.

Let $A = \{a_1, a_2, ..., a_l\}$. Define $A^0 := \phi$ and $A^i := \bigcup_{j=1}^{i} \{a_j\}$ for $i = 1, ..., l$. The submodularity of z enables us to assume without loss of generality that the elements of A are labelled such that $\rho_{A^{p-1}}(a_p) \geq \rho_{A^{q-1}}(a_q)$ for all p and q satisfying $1 \leq p \leq q \leq l$. Define $r := \lfloor l/d \rfloor d$. A^r can be partitioned into $\lfloor l/d \rfloor$ mutually exclusive subsets each of size d. By submodularity,

$$z(A^r) \leq \lfloor l/d \rfloor b.$$

Note that

$$z(A^r) = \sum_{i=1}^{r} \rho_{A^{i-1}}(a_i) \text{ and}$$
$$z(A) = \sum_{i=1}^{l} \rho_{A^{i-1}}(a_i).$$

Given the ordering of the elements of A,

$$\frac{z(A^r)}{r} \geq \frac{z(A)}{l}.$$

Combining the inequalities yields the statement of the proposition. ■

Conforti and Cornuéjols [CoC84] introduced the concept of *total curvature,* α, of a set function, defined as:

$$\alpha := \max_{j \in N_*} \{ \frac{\rho_\phi(j) - \rho_{N \setminus j}(j)}{\rho_\phi(j)} \}$$

where $N_* = \{j \in N : \rho_\phi(j) > 0\}$ and $\rho(\cdot)$ is the marginal value function defined above. A function is affine if and only if $\alpha = 0$. For nondecreasing submodular functions $\alpha \in [0, 1]$.

An independence system can be characterized in a number of different ways. A complete characterization is provided by its *rank function* $r : 2^N \to \mathcal{Z}_+$ defined by $r(S) := \max\{|X| : X \subseteq S \text{ and } X \in \mathcal{F}\}$. Note that $S \in \mathcal{F}$ if and only if $r(S) = |S|$. It is well known result that the rank function of a matroid is normalized, nondecreasing, and submodular (see [Sch03]). We will refer to $r(N)$, the cardinality of the largest independent set, as the *rank* of the independence system, and denote it by K.

A more compact characterization is in terms of the rank and *girth* of the independence system. Girth is the cardinality of the smallest dependent set in the system, and will be denoted by $k + 1$. A *uniform matroid* is one that is defined by $\mathcal{F} = \{X \subseteq N : |X| \leq K\}$. Note that for uniform matroids $K = k$, but this relationship does not hold in general.

It is known that any independence system $(N, \mathcal{F})$ can be decomposed into a finite number of matroids; that is, there exists matroids $(N, \mathcal{F}^i)$ $i = 1, ..., p'$ such that $\mathcal{F} = \bigcap_{i=1}^{p'} \mathcal{F}^i$ (see [KoH78]). Therefore, another way of characterizing an independence system is in terms of the *minimum number*, p, *of matroids into which it can be decomposed.* It is not known, in general, how to find p efficiently.

A final characterization is given by the *rank quotient*, q, of an independence system defined as

$$q = \min_{S \subseteq N} \frac{r_l(S)}{r(S)}$$

where $r_l(S)$ is the *lower rank* of S defined as the cardinality of the smallest maximal independent set in S. Clearly, $0 \leq q \leq 1$ with $q = 1$ if and only if the independence system is a matroid. Korte and Hausmann [HKJ80] computed the rank quotient for a few simple independence systems. For example, the independence systems arising from the Symmetrical Travelling Salesman Problem or from the matching problem have $q \geq \frac{1}{2}$. However, in general, it is not known how to find q efficiently. The following result by Hausmann et al. [HKJ80] is useful in comparing bounds:

Proposition 6.3 *(Hausmann et al. [HKJ80])*

If an independence system $I := (N, \mathcal{F})$ of rank quotient q is the intersection of p matroids then $q \geq \frac{1}{p}$.

6.3 Applications

The generality of Problem (P) allows a very large variety of combinatorial problems to be cast as special cases. In fact, any problem that can be formulated as the maximization of a normalized nondecreasing submodular function, z, over *any* set system, $S = (N, \mathcal{F})$, falls within the framework of Problem (P). To see this, consider the independence system $I = (N, \mathcal{F}')$ defined by $\mathcal{F}' := \{A : A \subseteq B \text{ for some } B \in \mathcal{F}\}$. It follows from the nondecreasing assumption of z that $\max_{A \in \mathcal{F}} z(A) = \max_{A \in \mathcal{F}'} z(A)$. Therefore, Problem (P) includes essentially all combinatorial problems with linear nondecreasing objective functions. Additionally, it includes those with nondecreasing submodular functions. The reader is referred to [Fuj91] and [Nar97] which are specialized monographs on submodularity as well as standard combinatorial optimization books such as [NeW88] and [Sch03] for examples of theoretical and practical problems involving submodularity. We note that the maximum cut problem in directed and undirected graphs is not a special case of Problem (P) because the cut function, while submodular, is not necessarily nondecreasing. We also note a useful characterization of nondecreasing submodular functions equivalent to properties (P4) and (P6). A function z is nondecreasing submodular if and only if (see [NeW88] for a proof):

$$z(B) \leq z(A) + \sum_{j \in B \setminus A} \rho_A(\{j\}) \quad \forall A, B \subseteq N.$$

The recognition of the existence of a nondecreasing submodular objective function in a problem can be subtle, and may require a transformation of the feasible space.

In the rest of this section we discuss three practical applications of Problem (P) where the feasible set is a matroid.

6.3.1 Combinatorial Auctions

In forward combinatorial auctions, buyers can bid on bundles (sets) of one or more items simultaneously. Let M denote the set of items to be sold and let B denote the set of bidders. Each bidder b has a valuation function $v^b : 2^M \to \mathcal{R}$. By allowing bids on bundles, combinatorial auctions enable buyers to express the nonlinearities inherent in their valuations arising from substitution or complementarity effects among the items. A vast literature exists on designing and analyzing auction mechanisms in general, and combinatorial auctions mechanisms in particular. The reader is referred to [Kle99] and [KaP03] for an introduction to the literature. We focus attention here on the *Winner Determination Problem* (WDP), faced by the auctioneer, of how to allocate items among bidders in order to maximize revenue. Formally, WDP can be cast as follows:

$$z^* := \{\max \sum_{b \in B} v^b(M^b) : \{M^b\}_{b \in B} \text{ is a partition of } M\}. \qquad (WDP)$$

Define the ground set $N := \{(a, b) : a \in M, b \in B\}$. Define the collection of independent sets, $\mathcal{F}$, to be those sets where no two elements correspond to the same item; that is, no item is assigned to multiple bidders. Thus a one-to-one correspondence exists between independent sets and (partial) feasible assignments. It can be verified that the collection of independent sets defines a matroid (whose rank is $|M|$). The valuation functions can be defined on N instead of on M. These functions are for most practical purposes normalized and nondecreasing. Some researchers have analyzed the case where valuation functions are additive in the items sold up to a budget constraint. Such functions are submodular. Lehmann et al. [LLN02] have also observed that substitution effects among items correspond precisely to the notion of submodular valuation functions. In either of these cases the functions v^b (and their analogues defined on the ground set N) are normalized, nondecreasing, and submodular for each bidder b. It then follows that the objective function in Problem (WDP) also inherits these properties (submodularity is preserved under addition). Hence, (WDP) is a special case of Problem (P).

We conclude this section by highlighting a result by Lehman et al. [LLN02] that is relevant to the bounds discussed in this part of the thesis. Note that sequential single-item auctions essentially provide a greedy approach to solving Problem (WDP)[1].

Proposition 6.4 *(Lehman et al. [LLN02])*

For normalized, non-decreasing, and submodular valuation functions, replacing a fully combinatorial auction by a sequence of single-item auctions produces total revenue that is at least half the optimal.

6.3.2 Bank Account Location

The bank account location problem analyzed in Cornuéjols et al. [CFN77] provided much of the initial motivation behind the study of Problem (P). Consider a firm wishing to maximize its funds by maintaining a fixed number of bank accounts in strategically located cities and assigning bill payments to accounts in such a way as to maximize the clearing time of payments. Formally, given an $|N| \times |N|$ nonnegative matrix $[c_{ij}]$ (representing, say, the interest saved by paying client i from an account located in city j), the problem is

$$\max_S \{\sum_{i \in N} \max_{j \in S} c_{ij} : S \subseteq N \text{ and } |S| \leq K\} \tag{6.2}$$

[1]A careful reading of the proofs of STANDARD_GREEDY in the literature shows that the performance guarantee of 2 applies in the special case of the auctions example, independent of item sequence. It suffices to select a bidder with the highest marginal valuation for each item, regardless of the ordering of items.

where K is the maximum number of accounts to open. It can easily be verified that the feasible set defines a (uniform) matroid, and that the objective function is submodular. The problem is NP-complete by reduction from the node covering problem. This problem can be viewed as a maximization version of the K-median problem defined by

$$\min_{S}\{\sum_{i\in N}\min_{j\in S} c_{ij} : S \neq \phi, S \subseteq N \text{ and } |S| \leq K\}. \tag{6.3}$$

It should be noted, however, that the K-median problem is not a special case of Problem (P) because we do not know of a transformation where assumptions (P4), (P5), and (P6) are simultaneously satisfied.

For examples of other problems having similar structure and arising in different applications, the reader is referred to [KPR98] and [DoK93].

6.3.3 Channel Assignment in Cellular Networks

A third application of Problem (P) arises in cellular networks and falls under the more generic problem of graph coloring.

Simon [Sim89] describes a setting where channels from a give set is to be assigned to users in a cellular radio network. Users in geographical proximity must be assigned different channels in order to prevent co-channel interference. The problem is to maximize the number of satisfied call requests. Simon describes an abstraction of this problem where each user is represented by a node in an undirected graph and edges represent pairs of nodes that must be assigned different channels (colors). The objective is to color the maximum number of nodes from a finite number of colors such that no two adjacent nodes have the same color. Building on earlier work by Johnson [Joh74], Simon considers an algorithm that starts by picking a any color and assigning it to the maximum number of nodes through solving a maximum stable set problem[2]. The colored nodes are then deleted and the same process is applied to the residual graph using the remaining colors. At each step the maximum stable set problem can be solved polynomially within ϵ of optimality

[2]The term 'stable set' of a graph is used here instead of the equivalent term 'independent set' of a graph to avoid confusion with independent sets of matroids.

using an approximation scheme developed by Hochbaum and Maass [HoM85] that exploits the special structure of the problem. Simon shows that the sequential coloring algorithm has performance guarantee of $e/(e-1)$.

This problem can be cast in the form of Problem (P) as follows. Consider a complete bipartite graph $G = (L \cup R, E)$ where L is the set of colors and R is the set of all stable sets in the original problem (that is, subsets of nodes where no pair shares an edge). Define a uniform matroid where the ground set $N := E$, and any subset of E of cardinality $|L|$ or less is feasible. Also define a set function $z : 2^E \rightarrow \mathcal{R}$ where for any $E' \subseteq E$, $z(E')$ is the number of nodes in the union of stable sets incident on the edges of E'; that is, $z(E') = \bigcup_{\{L_e, R_e\} \in E'} R_e$. It can be verified that:

- z is normalized, nondecreasing, and submodular;
- An optimal solution exists where the colors are assigned to mutually exclusive stable sets. Thus the matroid problem is equivalent to the coloring problem;
- A greedy solution of the matroid problem where ties are broken in favor of mutually exlusive stable sets is exactly equivalent to the sequential coloring algorithm proposed in [Sim89].

In Chapter 7, we show that the $e/(e-1)$ performance guarantee applies to the more general problem of maximizing normalized nondecreasing submodular functions over uniform matroids using Standard Greedy.

6.4 Standard Greedy Algorithm

A simple approach to finding a feasible solution to Problem (P) is the following standard best-in greedy algorithm:

Algorithm 6.1 Standard Greedy

Initialize $i := 0$; $G^0 := \phi$;

While G^i *is not maximal do*

$\{$ *Set* $i := i + 1$;

Select $x_i = \arg\max \{\rho_{G^{i-1}}(x) : x \in N \backslash G^{i-1}$ *and* $G^{i-1} \cup x \in \mathcal{F}\}$;

Set $G^i := G^{i-1} \cup x_i$. $\}$.

At each iteration i, the algorithm augments the incumbent solution G^{i-1} with an element x_i that yields maximum marginal objective function improvement while maintaining feasibility. The algorithm terminates once a maximal solution is constructed. Note that we consider maximal solutions only because the objective function is assumed nondecreasing.

It is well-known that STANDARD GREEDY is optimal when the objective function is affine and the feasible set is defined by a matroid (see Section 7.2.1). It is also easy to see that STANDARD GREEDY can produce arbitrarily bad solutions when the feasible set is defined by a general independence system even if the objective function is affine. The example below illustrates the myopic nature of STANDARD GREEDY leading it to produce suboptimal solutions when the objective function is submodular even when the feasible set is defined by a matroid.

Example 6.2

$N = \{e_1, e_2, e_3\}$. *The function* z *is defined as follows:*

A	ϕ	$\{e_1\}$	$\{e_2\}$	$\{e_3\}$	$\{e_1, e_2\}$	$\{e_1, e_3\}$	$\{e_2, e_3\}$	N
$z(A)$	0	1	1	1	1	2	2	2

It can be verified that this function is normalized, nondecreasing, and submodular. Now define the matroid $I := (N, \mathcal{F})$ *by*

$$\mathcal{F} = \{\phi, \{e_1\}, \{e_2\}, \{e_3\}, \{e_1, e_2\}, \{e_2, e_3\}\}.$$

It can be verified that this indeed defines a matroid. Now consider an application of STANDARD GREEDY *where the first element selected is* e_1 *(infinitesimal perturbations can be introduced to make this element the unique choice by the algorithm). In this case, the solution produced is* $\{e_1, e_2\}$ *of value* 1 *whereas the optimal value is* 2 *corresponding to the solution* $\{e_2, e_3\}$.

We review known results on the performance of STANDARD GREEDY in the next chapter. It will be shown that the performance ratio of 2 produced in the above example is the worst possible for Problem (P) when I is a matroid.

Chapter 7

Literature Review

In this chapter we review the literature on complexity results pertaining to Problem (P) and performance bounds applicable to greedy type algorithms.

It is useful to distinguish between the following classes of objective functions and independence systems:

$\mathcal{Z}_L$: set of all linear functions; that is, submodular functions that can be expressed as $z(A) = \sum_{a \in A} c(a)$ for all $A \subseteq N$ for some ground set N and arbitrary scalars $c(a)$ and $a \in N$. The independence assumption allows us to assume without loss of generality that $c(a) \geq 0$ for all $a \in N$;

$\mathcal{Z}_S$: set of all nondecreasing submodular functions;

$\mathcal{F}_U$: set of all uniform matroids; $\mathcal{F}_U = \{\mathcal{F} : \mathcal{F} = \{X \subseteq N : |X| \leq K\}$ for some $K \in \mathcal{Z}_+\}$ for some ground set N;

$\mathcal{F}_M$: set of all matroids; $\mathcal{F}_M = \{\mathcal{F} : (N, \mathcal{F})$ is a matroid$\}$ for some ground set N;

$\mathcal{F}_K$: the set consisting of all sets that are feasible solutions to a knapsack constraint on N; $\mathcal{F}_K = \{\mathcal{F} : \mathcal{F} = \{X \subseteq N : \sum\limits_{x \in X} w(x) \leq B\}$ for some $B \in \mathcal{Z}_+$ and $w(x) \in \mathcal{Z}_+ \ \forall x \in N\}$;

$\mathcal{F}_I$: set of all independence systems; $\mathcal{F}_I = \{\mathcal{F} : (N, \mathcal{F})$ is an independence system$\}$.

A hierarchy of these classes of independence systems is depicted in Figure 7-1.

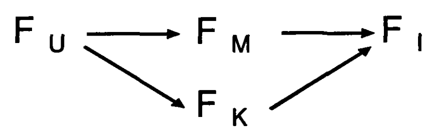

Figure 7-1: A hierarchy of the different classes of feasible sets. $\mathcal{F}1 \longrightarrow \mathcal{F}2$ indicates that $\mathcal{F}1 \subset \mathcal{F}2$.

	Z_L	Z_S
$\mathcal{F}_U$	Polynomial	NP-complete
$\mathcal{F}_M$	Polynomial	NP-complete
$\mathcal{F}_K$	NP-complete	NP-complete
$\mathcal{F}_I$	NP-complete	NP-complete

Table 7.1: Complexity results for Problem (P)

We use the short hand notation $A|B$ to denote Problem (P) with the objective function restricted to class A and the feasible set restricted to class B. For example $\mathcal{Z}_L|\mathcal{F}_I$ refers to the maximization of linear functions over independence systems.

7.1 Complexity Results

Table 7.1 summarizes the complexity of Problem (P). The optimality of STANDARD GREEDY for $Z_L|\mathcal{F}_M$ was shown by Gale [Gal68] and Edmonds [Edm71]. Karp [Kar72] showed the $NP-$*completeness* of the node covering and knapsack problems which are special cases of $Z_S|\mathcal{F}_U$ and $Z_L|\mathcal{F}_K$, respectively. The remaining entries in Table 7.1 follow by the implications in Figure 7-1 and the fact that $Z_L \subseteq Z_S$.

7.2 Performance Bounds

The results summarized in Figure 7-1 and Table 7.1 have motivated the development of approximation algorithms and the analysis of their worst-case performance. Let z^a denote the value of the solution obtained by the approximation algorithm under consideration and let z^* denote the optimal value. An algorithm has an approximation guarantee of $\eta \geq 1$ if $z^*/z^a \leq \eta$ for all instances of Problem (P). A summary of the main approximation

	Z_L		Z_S		
$\mathcal{F}_U$	1		$[1-[\frac{K-1}{K}]^K]^{-1} < \frac{e}{e-1}$		[NWF78]
$\mathcal{F}_M$	1	[Edm71]	2		[FNW78]
			$1+\alpha$		[CoC84]
			$[1-\alpha_G\frac{K-1}{K}]^{-1}$		[CoC84]
			2	$K \geq 2k$	[CoC84]
			$[1-\frac{k}{K}[\frac{K-1}{K}]^{2k-K}]^{-1}$	$K \leq 2k$	
$\mathcal{F}_K$	none		none		
$\mathcal{F}_I$	p	[KoH78]	$p+1$		[FNW78]
	$\frac{1}{q}$	[Jen76]	$[1-[\frac{K-1}{K}]^k]^{-1}$		[FNW78]
			$p+\alpha$		[CoC84]
			$\alpha[1-[\frac{K-\alpha}{K}]^k]^{-1}$		[CoC84]

Table 7.2: Approximation guarantees for Standard Greedy when applied to Problem (P)

guarantees of STANDARD GREEDY when applied to Problem (P) is given in Table 7.2 and discussed below. The parameters α, q, p, and K are as defined in Section 6.2. The parameter α_G is defined below. In many cases, these bounds are sharp, which roughly means that instances achieving the given bound exist. For the exact meaning of sharpness for a particular bound the reader can consult the original reference.

7.2.1 Linear Functions

The optimality of STANDARD GREEDY for $Z_L|\mathcal{F}_M$ was shown by Edmonds [Edm71]. The reverse result characterizing matroids as the only set systems where STANDARD GREEDY guarantees an optimal solution under any objective function $z \in Z_L$ was shown by Rado [Rad57]. Both results were also proven indirectly by Gale [Gal68].

In contrast, STANDARD GREEDY does not have a constant approximation ratio when applied to $Z_L|\mathcal{F}_K$. To see this consider the following example. For some integer $s > 1$, let $N = \{e_0, e_1, ..., e_s\}$, $w_0 = s$, $w_i = 1$ for $i = 1, ..., s$, $c_0 = 1+\epsilon$ for some $\epsilon > 0$, $c_i = 1$ for $i = 1, ..., s$, and $B = s$. The objective is $\max\limits_{X \subseteq \{0,1,...,s\}} \{\sum_{x\in X} c_x : \sum_{x \in X} w_x \leq B\}$. The optimal solution is $X = \{1, ..., s\}$ with $z^* = s$. However STANDARD GREEDY selects $X = \{0\}$ with $z^G = 1+\epsilon$. Even a simple modification of the STANDARD GREEDY, where at each iteration the item with highest ratio of objective function coefficient to weight is selected, will not work. Consider the following example (adapted from [KMN99]): $N = \{e_1, e_2\}$, $w_1 = 1$, $w_2 = s$, $c_1 = 1$, $c_2 = s - 1$, and $B = s$. The optimal solution is $X = \{2\}$ with $z^* = s - 1$. The modified Greedy algorithm will yield $z^G = 1$ because element e_1 has the highest value to weight ratio. The approximation guarantee can be made arbitrarily large by selecting s arbitrarily large. We note that this property of STANDARD GREEDY is in sharp contrast to the *fully polynomial time approximation scheme* (FPTAS) proposed by Ibarra and Kim for $Z_L|\mathcal{F}_K$ [IbK75]. FPTAS means that for any $\epsilon > 0$, the approximation algorithm has running time that is polynomial in the instance size and in $\frac{1}{\epsilon}$, and is guaranteed to produce a solution whose relative error is within ϵ. In a sense, FPTAS is almost the best any algorithm can achieve for $NP - Hard$ problems unless $P = NP$.

The performance of STANDARD GREEDY to $Z_L|\mathcal{F}_I$ has been studied by a number of researchers. For example, Korte and Hausmann [KoH78] have shown that the performance of STANDARD GREEDY is bounded from above by the minimum number p of matroids into which the independence system can be decomposed. Fisher et al. [FNW78] consider special cases of Problem (P) that can be formulated as linear integer programs. They prove a stronger performance result where the relative error of the greedy solution compared to that of the *LP relaxation* of Problem (P) is bounded from above by p. Jenkyns [Jen76] has shown that the performance ratio is upper bounded by $\frac{1}{q}$, where q is the rank quotient of the independence system. Recall that Hausmann et al. [HKJ80] have shown that $\frac{1}{q} \leq p$ for any independence system.

7.2.2 Nondecreasing Submodular Functions

The performance of STANDARD GREEDY when applied to $Z_S|\mathcal{F}_U$ has been analyzed in detail by Nemhauser et al. [NWF78]. They derive a sharp bound of $[1 - [\frac{K-1}{K}]^K]^{-1} < \frac{e}{e-1}$ where K is the rank of the matroid. For $Z_S|\mathcal{F}_M$, a sharp approximation ratio of 2 was obtained by Fisher, Nemhauser and Wolsey [FNW78]. They also derive a bound of $[1 - [\frac{K-1}{K}]^k]^{-1}$ where $k + 1$ is the girth of the matroid. Conforti and Cornuéjols [CoC84] generalize the bounds of [FNW78] to $1 + \alpha$ and $[\frac{1}{\alpha}[1 - [\frac{K-\alpha}{K}]^k]]^{-1}$ where α is the total curvature of the submodular function defined in Section (6.2). Conforti and Cornuéjols [CoC84] also derive two other bounds:

- $\eta \leq [1 - \alpha_G[\frac{K-1}{K}]]^{-1}$ where α_G is the *greedy curvature* of the objective function defined as $\max_{1 \leq i \leq T-1} \min_{j \in N^i} \{\frac{\rho_\phi(\{j\}) - \rho_{G^{i-1}}(\{j\})}{\rho_\phi(\{j\})}\}$ where $N^i := \{j \in N \backslash G^{i-1} : G^i \cup \{j\} \in \mathcal{F}$ and $\rho_\phi(\{j\}) > 0\}$ and T is the number of iterations executed by STANDARD GREEDY; and

- $\eta \leq [1 - \frac{k}{K}[\frac{K-1}{K}]^{2k-K}]^{-1}$ if $K \leq 2k$ and $\eta \leq 2$ otherwise. This bound is sharp.

For $Z_S|\mathcal{F}_I$, Fisher et al. [FNW78] establish the bounds $p+1$ and $[1-[\frac{K-\alpha}{K}]^k]^{-1}$. These were generalized to $p+\alpha$ and $[\frac{1}{\alpha}[1-[\frac{K-\alpha}{K}]^k]]^{-1}$ by Conforti and Cornuéjols [CoC84].

7.2.3 Variants of Standard Greedy

A variant of STANDARD GREEDY that augments the current solution by a subset of *constant* size has been analyzed for two special cases of Problem (P). We refer to such an algorithm as R-GREEDY, where R is the constant augmentation size. Suppose $K = mR - r$ for some integer m, and let $\lambda = (R-r)/R$. Nemhauser et al. [NWF78] have shown that for $Z_S|\mathcal{F}_U$ R-GREEDY has an approximation guarantee of $[1 - (\frac{m-\lambda}{m})(\frac{m-1}{m})^{m-1}]^{-1}$. This bound is sharp if $r = 0$. For $Z_L|\mathcal{F}_I$, Hausmann et al. [HKJ80] have shown that the approximation guarantee q still holds even though R-GREEDY may produce a solution of lower value than STANDARD GREEDY. This bound is, in general, not sharp. Note that these two bounds consider two extremes of Problem (P) where the feasible set, respectively, the objective function, has a simple form. Our analysis in this part of the thesis applies to Problem (P) in general with augmentations of arbitrary variable size, and results in stronger bounds.

Another variant studied by Nemhauser at al. [NWF78] for $Z_S|\mathcal{F}_U$ involves the enumeration of all subsets of cardinality c, where $c \in \{1, 2, ..., K-1\}$, followed by completing each of the $\binom{n}{c}$ partial solutions using a standard greedy approach until a maximal solution is reached. The best solution produced is then selected. The authors have shown that such an algorithm has an approximation guarantee of

$$\left[1 - \left(\frac{K-c}{K}\right)\left(\frac{K-c-1}{K-c}\right)^{K-c}\right]^{-1}.$$

7.3 Hardness of Approximation

We conclude this section by mentioning an important inapproximability result. Fiege [Fei78] has shown that the best possible approximation guarantee for the max K-cover problem is $\frac{e}{e-1}$ assuming $P \neq NP$. Max K-cover is the problem of selecting at most K subsets whose union has maximum cardinality from a collection of subsets of a given ground set. Max K-cover is a special case of $Z_S|\mathcal{F}_U$. This implies that STANDARD GREEDY achieves an approximation guarantee for $Z_S|\mathcal{F}_U$ that is best possible unless $P = NP$.

A somewhat related result, that we mention here for completeness, is the following one by Nemhauser and Wolsey [NeW78]:

Proposition 7.1 *(Nemhauser and Wolsey [NeW78])*

For the $Z_S|\mathcal{F}_U$ family of problems, no algorithm requiring a number of function evaluations polynomial in n (the cardinality of the ground set) has an approximation guarantee better than $\frac{e}{e-1}$.

It is important to note that this result measures computation in terms of n rather than in terms of the instance input size which could be exponential in n for complicated objective functions.

Chapter 8

Generalized Greedy Algorithm

A main question addressed in this part of the dissertation is the following: *What is the performance of a generalization of* STANDARD GREEDY *where at each iteration the incumbent solution is augmented, not by a single element but instead, by a subset of elements of arbitrary cardinality*? At one extreme this algorithm is the same as standard greedy. At the other extreme it is synonymous with complete enumeration. This yields a family of algorithms that represents a continuum of trade-offs between optimality and tractability. We are interested in deriving performance guarantees for such an algorithm in terms of its parameters.

The generalized algorithm is formally stated below, followed by a few explanatory remarks.

Algorithm 8.1 GENERALIZED GREEDY

Initialize $i := 0$; $G^0 := \phi$;

While G^i *is not maximal do*

{ *Set* $i := i + 1$;

Select $m_i, n'_i \in \mathcal{Z}_+$;

Select $\gamma_i \geq \gamma'_i := \max\{\rho_{G^{i-1}}(X) : X \subseteq N \text{ and } |X| \leq m_i\}$;

Select $X_i = \arg\max \{\rho_{G^{i-1}}(X) : G^{i-1} \cup X \in \mathcal{F} \text{ and } |X| \leq n'_i\}$;

Set $G^i := G^{i-1} \cup X_i$. }.

At each iteration i, the algorithm performs two steps. The first is an *unconstrained* *optimization* step yielding an upper bound, γ_i, on the largest unconstrained marginal improvement, γ_i', for an augmentation of size at most m_i. This is followed by a *constrained optimization* step to determine an augmentation X_i of size at most n_i' that yields a new feasible solution of maximum marginal improvement. The actual augmentation size, $|X_i|$, will be denoted by n_i. The short-hand notation GREEDY$(\mathbf{m}, \mathbf{n})$ will be used to denote an implementation of GENERALIZED GREEDY with $\mathbf{m} := [m_i]$ and $\mathbf{n} := [n_i]$ as the step sizes chosen. STANDARD GREEDY is equivalent to GREEDY$(\mathbf{m}, \mathbf{1})$ with a sufficiently large value for γ_i assumed at each iteration. In the analysis below, t denotes the number of iterations executed by GENERALIZED GREEDY and $G := G^t$ denotes the final solution constructed with value $z^G := z(G)$. For each $i \in \{1, ..., t\}$, define $\delta_i := \rho_{G^{i-1}}(X_i)$ and $\lambda_i := \delta_i - \rho_{N \setminus X_i}(X_i)$. δ_i is the constrained marginal improvement at iteration i and λ_i is a modified marginal improvement value that is useful in the analysis. Finally, average values are denoted by $\overline{\delta}_i := \delta_i / n_i$ and $\overline{\gamma}_i := \gamma_i / m_i$ for $i = 1, ..., t$.

It should be noted that while the unconstrained optimization step has no effect on the construction of the greedy solution, it does play an important role in gathering information about the objective function in the vicinity of the solution constructed. This information in general improves the quality of the bounds computed. The incorporation of this information will be evident in the bounds developed in Chapter 9. Therefore, the purpose of the algorithm can be characterized as providing a primal solution *as well as* dual information that leads to an improved bound on the optimality gap. Intuitively, the unconstrained step helps distinguish between cases where the best marginal improvement attainable is small because the objective function is flat in the vicinity of the current solution and cases where is it small because of restrictions imposed by the feasible set $\mathcal{F}$. The next example provides an illustration.

Example 8.1

Consider an instance of Problem (P) where $N = \{e_1, e_2, e_3\}$ *and* I *is a matroid of rank* 2. *Suppose* GENERALIZED GREEDY *is implemented with unit step sizes; that is the algorithm consists of two iterations with* $n_1' = n_2' = m_1 = m_2 = 1$. *Let the marginal values* *obtained be as follows:*

i	1	2
δ_i	1	0
γ_i	1	0

The value of the greedy solution obtained equals $\delta_1 + \delta_2 = 1$. *Based on the constrained marginal values alone (the* δ_i *values) it is possible that the optimal value is as high as* 2 *(see Example 6.2). However, given knowledge of the unconstrained marginal values (the* γ_i *values), the optimal value cannot exceed* 1. *To see this, assume that the element selected by the algorithm in the first iteration is* e_1. *The fact that* $\gamma_2 = 0$ *implies that* $z(\{e_1, e_2\}) = z(\{e_1, e_3\}) = z(\{e_1\}) = 1$. *Submodularity implies that* $z(\{e_1, e_2, e_3\}) \leq z(\{e_1, e_2\}) + z(\{e_1, e_3\}) - z(\{e_1\}) = 1$. *This implies that* $z(\{e_2, e_3\}) \leq z(\{e_1, e_2, e_3\}) \leq 1$ *by the nondecreasing assumption. Therefore the unconstrained optimization step has enabled us to establish the optimality of the greedy solution in this case.*

We conclude this section by stating a monotonicity property of the average best constrained and unconstrained marginal improvements of Generalized Greedy. This is a direct consequence of the submodularity of the objective function.

Proposition 8.1

Any application of Generalized Greedy *to Problem (P) yields:*

$$\begin{aligned} \overline{\delta}_1 &\geq \overline{\delta}_2 \geq \cdots \geq \overline{\delta}_t; \\ \overline{\gamma}'_1 &\geq \overline{\gamma}'_2 \geq \cdots \geq \overline{\gamma}'_t. \end{aligned}$$

Proof.

We prove the monotonicity of the $\{\overline{\delta}_i\}_{i=1}^t$ sequence only. The proof for $\{\overline{\gamma}'_i\}_{i=1}^t$ is almost identical.

Fix $i \in \{1, ..., t-1\}$. By definition, $\overline{\delta}_i = \rho_{G^{i-1}}(X_i)/n_i$ and $\overline{\delta}_{i+1} = \rho_{G^i}(X_{i+1})/n_{i+1}$. We consider three cases:

- $n_i = n_{i+1}$. By independence, X_{i+1} is a candidate for augmentation at iteration i.

Therefore, $\rho_{G^{i-1}}(X_i) \geq \rho_{G^{i-1}}(X_{i+1})$. By submodularity, $\rho_{G^{i-1}}(X_{i+1}) \geq \rho_{G^i}(X_{i+1})$. Therefore $\overline{\delta}_i \geq \overline{\delta}_{i+1}$.

- $n_i < n_{i+1}$. By independence, each subset of X_{i+1} of cardinality n_i is a candidate for augmentation at iteration i. Propositions 6.1 and 6.2 imply $\rho_{G^{i-1}}(X_{i+1}) \leq \frac{n_{i+1}}{n_i}\rho_{G^{i-1}}(X_i)$. By submodularity, $\rho_{G^i}(X_{i+1}) \leq \rho_{G^{i-1}}(X_{i+1})$. Therefore, $\overline{\delta}_i \geq \overline{\delta}_{i+1}$.

- $n_i > n_{i+1}$. By Proposition 6.2, there exists $A \subseteq X_i$ such that $|A| = n_i - n_{i+1}$ and $\rho_{G^{i-1}}(A) \geq \frac{n_i - n_{i+1}}{n_i}\rho_{G^{i-1}}(X_i)$. By independence, $A \cup X_{i+1}$ is a candidate for augmentation at iteration i. Therefore, $\rho_{G^{i-1}}(X_i) \geq \rho_{G^{i-1}}(A \cup X_{i+1}) = \rho_{G^{i-1}}(A) + \rho_{G^{i-1} \cup A}(X_{i+1})$. By submodularity, $\rho_{G^i}(X_{i+1}) \leq \rho_{G^{i-1} \cup A}(X_{i+1})$. Combining these inequalities yields $\rho_{G^i}(X_{i+1}) \leq \rho_{G^{i-1}}(X_i) - \rho_{G^{i-1}}(A) \leq \rho_{G^{i-1}}(X_i) - \frac{n_i - n_{i+1}}{n_i}\rho_{G^{i-1}}(X_i) = \frac{n_{i+1}}{n_i}\rho_{G^{i-1}}(X_i)$. Therefore, $\overline{\delta}_i \geq \overline{\delta}_{i+1}$.

■

Chapter 9

Analysis of Bounds for Matroids

In this chapter we develop performance bounds on GENERALIZED GREEDY when applied to Problem (P).

We first pose the question of whether bounds on the performance of GENERALIZED GREEDY can be obtained by applying the results for STANDARD GREEDY to a *transformed* problem. More specifically, given Problem (P) and step-size vectors $\mathbf{m}$ and $\mathbf{n}$, can we construct a modified independence system $I' = (N', \mathcal{F}')$ and a normalized nondecreasing submodular function $z' : 2^{N'} \to \mathcal{R}$ such that:

1. $z^* := \max\{z(S) : S \in \mathcal{F}\} = \max\{z'(S) : S \in \mathcal{F}'\}$;
2. If G is the solution obtained by applying GREEDY$(\mathbf{m}, \mathbf{n})$ to Problem (P) and G' is the solution obtained by applying STANDARD GREEDY to the transformed problem $\max\{z'(S) : S \in \mathcal{F}'\}$ then $z(G) = z'(G')$.

Such a transformation, if it exists, would deliver an elegant procedure for extending the applicability of the bounds reviewed in Section 7 to the more general setting of GENERALIZED GREEDY. Consider the extreme case where $I = (N, \mathcal{F})$ is a matroid of rank K and GENERALIZED GREEDY consists of a single augmentation of size K. In this special case, the following transformation would satisfy conditions 1 and 2 above: i) define N' to be the set obtained by enumerating all subsets of N of cardinality K, ii) define $\mathcal{F}'$ to consist of all singletons corresponding to bases of I, and iii) define $z'(A) := z(\bigcup_{e \in A}\{N_e\})$ where

$N_e \subseteq N$ is the set of elements represented by $e \in N'$. In general, it would seem reasonable to create N' by enumerating all subsets corresponding to all possible step sizes in $\mathbf{m}$ and $\mathbf{n}$. However, we argue below that such a reduction approach does not *seem* to work for the following reasons:

- The transformation $\mathcal{F} \to \mathcal{F}'$ is not uniquely defined. Each basis in $\mathcal{F}$ would correspond to a number of alternative subsets of N', and it is not clear which is the right representation. In particular, if I is a matroid, I' might not be a matroid leading to an inferior set of bounds.

- While the objective function transformation $z'(A) := z(\bigcup_{e \in A}\{N_e\})$ yields a function $z' : 2^{N'} \to \mathcal{R}$ that can easily be shown to inherit the normalization, monotonicity, and submodularity properties of z, the reverse is not true in general. Thus, assuming these properties on z' is tantamount to assuming a more general class of objective functions than that assumed by properties (P4), (P5), and (P6); that is, we would be solving a harder problem. It is arguable, however, that a relaxation of these properties is expected because GENERALIZED GREEDY 'washes away' the need for these properties *within* an augmentation.

Unable to find a transformation that works, we will derive the bounds in this section from scratch. Our derivation, however, will build on ideas used in [NWF78], [FNW78], and [CoC84] to analyze bounds on STANDARD GREEDY.

9.1 Comparison of Bounds

We mentioned towards the end of Section 7 that the bounds derived here are stronger than those available in the literature. In this subsection, we make notions of *sharpness* and *strength* of bounds more precise.

Consider any maximization problem $\mathcal{P}$ defined by a set of instances. For an instance $p \in \mathcal{P}$, let $z^*(p)$ denote its optimal value. Suppose we have a deterministic algorithm A that terminates with a feasible solution to problem $\mathcal{P}$. Let $z^A(p)$ denote the solution obtained by A when applied to instance $p \in \mathcal{P}$. The performance of the algorithm in this instance is the ratio $\frac{z^*(p)}{z^A(p)}$. Let $b^i : \mathcal{P} \to \mathcal{R}, i = 1, 2$, be two upper bound *functions* on the performance ratio.

- A bound b^i is *sharp* if $b^i(p) = \lim_{i\to\infty} \frac{z^*(p_i)}{z^A(p_i)}$ for some sequence $\{p_i\} \subseteq \mathcal{P}$.
- Bound b^1 is *stronger* than bound b^2 if:
  1. $b^1(p) \leq b^2(p)$ for all $p \in \mathcal{P}$;
  2. $b^1(p) < b^2(p)$ for some $p \in \mathcal{P}$; and
  3. the time required to compute b^1 is polynomial in the time required to compute b^2 for each $p \in \mathcal{P}$.
- A constant factor bound b^1 is *best possible unless* $P = NP$ if no polynomial algorithm can achieve a better worst-case constant factor approximation assuming $P \neq NP$.

From the bounds reviewed in Chapter 7, it should be clear that the $1 + \alpha$ bound of Conforti and Cornuéjols [CoC84] is stronger than the 2 bound of Fisher et al. [FNW78]. In general, one should be careful in making claims about strength. Often the final bound derived by a researcher is a simplification of more complicated intermediate expressions in their proof that are themselves valid bounds. It is important not to make claims about strength by comparing final results rather than intermediate ones that were deliberately simplified. We claim that the bounds developed in the next section are truly stronger than many reported in Table (7.2).

9.2 Matroid Bound

In this section we assume that $I = (N, \mathcal{F})$ is a matroid and derive our main performance bound for Generalized Greedy when applied to matroids.

Before presenting a formal statement of the bound, we first attempt an 'intuitive' explanation of why a greedy type algorithm could be expected to perform 'reasonably well' under the matroid assumption. An obvious weakness of a greedy approach is that intermediate solutions are not revised, they are simply augmented. Any greedy algorithm thus runs the risk of getting locked in a partial solution that offers very small potential for improvement.

The advantage of submodularity, however, is that the largest gains are achieved at the beginning (Proposition 8.1), precisely when the algorithm is constrained the least. Contrast this with a supermodular function where the entire value of a set could be due to the final element added. Submodularity reduces the need to look far ahead.

A second explanation is that the matroid assumption, in particular the growth property (P3), ensures a high level of 'connectivity' among the feasible sets. Put differently, a partial greedy solution can still incorporate elements from the best basis. This helps prevent intermediate solutions from getting completely locked into a bad neighborhood. This is evident in a number of 'exchange / replacement' type results developed for matroids that we state here and use in the proof of the performance bound. The following *exchange* result was proven by Brylawski [Bry73], Greene [Gre73] and Woodall [Woo74].

Proposition 9.1 *(Brylawski [Bry73], Greene [Gre73], and Woodall [Woo74])*

Let A and B be two bases of a matroid. If A is partitioned into A_1 and A_2 then B can be partitioned into B_1 and B_2 such that $A_1 \cup B_2$ and $A_2 \cup B_1$ are bases.

The following *replacement* result is due to Greene and Magnanti [GrM75].

Proposition 9.2 *(Greene and Magnanti [GrM75])*

Let A and B be two bases of a matroid. If A is partitioned into $\{A_1, A_2, ..., A_l\}$ then B can be partitioned into $\{B_1, B_2, ..., B_l\}$ such that $(A \backslash A_i) \cup B_i$ is a base for each $i = 1, ..., l$.

The following observation is a consequence of property (P3$'$) for matroids.

Proposition 9.3

In Propositions 9.1 and 9.2, we necessarily have $|A_i| = |B_i|$ for all i.

Proof.

First consider Proposition 9.1. Property (P3$'$) implies that

$$|A_1 \cup B_2| = |A_2 \cup B_1| = K,$$

where K is the matroid rank. Moreover, the fact that (A_1, A_2) is a partition of A and (B_1, B_2) is a partition of B implies

$$\begin{aligned} K &= |A_1 \cup A_2| = |A_1| + |A_2|, \text{ and} \\ K &= |B_1 \cup B_2| = |B_1| + |B_2|. \end{aligned}$$

Clearly,

$$\begin{aligned} K &= |A_2 \cup B_1| \leq |A_2| + |B_1| = K - |A_1| + |B_1|, \text{ and} \\ K &= |A_1 \cup B_2| \leq |A_1| + |B_2| = K - |A_2| + |B_2|. \end{aligned}$$

By summing the above two inequalities it is clear that they must hold with equality. Therefore, $|A_1| = |B_1|$ and $|A_2| = |B_2|$.

Now consider the same argument for Proposition 9.2.

$$K = |(A \backslash A_i) \cup B_i| \leq |A \backslash A_i| + |B_i| = K - |A_i| + |B_i|, \qquad i = 1, ..., l$$

By summing the above l inequalities and noting that $\sum_{i=1}^{l} |A_i| = \sum_{i=1}^{l} |B_i|$, we conclude that each inequality must hold with equality, and thus $|A_i| = |B_i|$, $i = 1, ..., l$. ■

We now state the main result of this section. The reader is referred to Chapter 6 for definition of the notation used. The proof of this result builds on ideas used in [NWF78], [FNW78], and [CoC84] to analyze bounds for STANDARD GREEDY.

Proposition 9.4

Let $I = (N, \mathcal{F})$ be a matroid of rank K and let $z : 2^N \rightarrow \mathcal{R}$ be a normalized nondecreasing submodular function. Suppose GREEDY$(\mathbf{m}, \mathbf{n})$ *is applied to Problem (P). For each $i \in \{0, ..., t-1\}$, let L_i be any subset of $\{1, ..., i\}$[1]. Define $n^i_{L_i} := \sum_{j \in L_i} n_j$ and*

$$d^i_{L_i} := \begin{cases} m_{i+1} & \textit{if } |G^i| - m_{i+1} < n^i_{L_i} < |G^i|, \\ |G^i| - n^i_{L_i} & \textit{otherwise.} \end{cases}$$

[1] *We adopt the convention that $\{1, ..., i\} = \phi$ if $i = 0$.*

Then, the following bound on the optimal value z^* holds:

$$z^* \leq \min_{i \in \{0,...,t-1\}} [z(G^i) + (K - |G^i|)\,\overline{\delta}_{i+1} + d^i_{L_i}\,\overline{\gamma}_{i+1} + \sum_{j \in L_i} \lambda_j].$$

Proof.

Let S denote an optimal solution to Problem (P) that is maximal (with respect to inclusion). For $i = 0$, the bound holds by noting that $z^* = z(\phi) + \rho_\phi(S) \leq z(\phi) + (K/n_1)\,\delta_1$ by Proposition (6.2). Fix $i \in \{1, ..., t-1\}$. $\{G^i, G\backslash G^i\}$ defines a partition of G. By Proposition 9.1, there exists a partition $\{S_1, S_2\}$ of S such that $G^i \cup S_1$ is a base. $|S_1| = K - |G^i|$ and $|S_2| = |G^i|$.

Now we define a standard operation on matroids. Let $I := (N, \mathcal{F})$ be a matroid. For any $A \subseteq N$, the contraction of I with respect to A is the set system $I/A := (N\backslash A, \mathcal{F}')$ where $C \subseteq N\backslash A$ is an element of $\mathcal{F}'$ if and only if $C \cup B \in \mathcal{F}$ for some maximal set $B \subseteq A$ (maximal with respect to I). It can be shown that I/A is a matroid (see, for example, Schrijver [Sch03]).

Consider the matroid $I/S_1 := (N\backslash S_1, \mathcal{F}')$ obtained by contracting S_1. Clearly, G^i and S_2 are bases of I/S_1. Recall that X_j is the augmentation performed by $\textsc{Greedy}(\mathbf{m}, \mathbf{n})$ at iteration j. Given partition $\{X_1, ..., X_i\}$ of G^i, let $\{S_2^1, ..., S_2^i\}$ be a partition of S_2 satisfying the property stated in Proposition 9.2; that is, $(G^i\backslash X_j) \cup S_2^j \in \mathcal{F}'$ and $|X_j| = |S_2^j|$ for each $j \in \{1, ..., i\}$. Given L_i, define $S_2' := \bigcup_{j \in L_i} S_2^j$, and $G' := \bigcup_{j \in L_i} X_j$.

Clearly, $\{S_1, S_2\backslash S_2', S_2'\}$ is a partition of S. Therefore,

$$\begin{aligned} z(S) &= z(G^i \cup S) - \rho_S(G^i) \\ &\leq z(G^i) + \rho_{G^i}(S_1) + \rho_{G^i}(S_2\backslash S_2') + \rho_{G^i}(S_2') - \rho_S(G^i) \\ &\leq z(G^i) + \rho_{G^i}(S_1) + \rho_{G^i}(S_2\backslash S_2') + \rho_{G^i}(S_2') - \rho_S(G'), \end{aligned} \tag{9.1}$$

where the first inequality holds because z (or more precisely, ρ_{G^i}) is submodular and the second inequality holds because z is nondecreasing. We next bound the terms in the right-hand side of inequality (9.1).

Proposition 6.2 implies

$$\rho_{G^i}(S_1) \le \frac{|S_1|}{n_{i+1}} \delta_{i+1} = (K - |G^i|)\, \overline{\delta}_{i+1}. \tag{9.2}$$

To bound the term $\rho_{G^i}(S_2 \backslash S_2')$, we consider separately three cases:

Case i) $\sum_{j \in L_i} n_j = |G^i|$: Here, $|S_2 \backslash S_2'| = 0$. Therefore, $\rho_{G^i}(S_2 \backslash S_2') = 0 = d^i_{L_i}\, \overline{\gamma}_{i+1}$.

Case ii) $|G^i| - m_{i+1} < \sum_{j \in L_i} n_j < |G^i|$: Here, $0 < |S_2 \backslash S_2'| < m_{i+1}$. Therefore, $\rho_{G^i}(S_2 \backslash S_2') \le \gamma_{i+1} = d^i_{L_i}\, \overline{\gamma}_{i+1}$.

Case iii) $\sum_{j \in L_i} n_j \le |G^i| - m_{i+1}$. Here, $|S_2 \backslash S_2'| \ge m_{i+1}$. Therefore, $\rho_{G^i}(S_2 \backslash S_2') \le \frac{|S_2 \backslash S_2'|}{m_{i+1}} \gamma_{i+1} = (|G^i| - \sum_{j \in L_i} n_j)\, \overline{\gamma}_{i+1} = d^i_{L_i}\, \overline{\gamma}_{i+1}$.

Therefore in all three cases,

$$\rho_{G^i}(S_2 \backslash S_2') \le d^i_{L_i}\, \overline{\gamma}_{i+1}. \tag{9.3}$$

Finally, we bound the term $\rho_{G^i}(S_2') - \rho_S(G')$. Recall that $(G^i \backslash X_j) \cup S_2^j \in \mathcal{F}'$ and $|X_j| = |S_2^j|$ for each $j \in \{1, ..., i\}$. This implies that $S_1 \cup (G^i \backslash X_j) \cup S_2^j \in \mathcal{F}$ for each $j \in \{1, ..., i\}$. In particular, $G^{j-1} \cup S_2^j \in \mathcal{F}$ for each $j \in \{1, 2, ..., i\}$. Therefore S_2^j is a candidate for augmentation at iteration j. The operation of $\textsc{Greedy}(\mathbf{m}, \mathbf{n})$ guarantees that $\rho_{G^{j-1}}(S_2^j) \le \delta_j$. Denote $S_2^j \cap G^i$ by A^j. Therefore,

$$\begin{aligned}
\rho_{G^i}(S_2') - \rho_S(G') &\le \sum_{j \in L_i} \rho_{G^i}(S_2^j \backslash A^j) - \rho_S(G') \\
&\le \sum_{j \in L_i} \rho_{G^{j-1}}(S_2^j \backslash A^j) - \rho_S(G') \\
&= \sum_{j \in L_i} [\rho_{G^{j-1}}(S_2^j) - \rho_{G^{j-1} \cup (S_2^j \backslash A^j)}(A^j)] - \rho_S(G') \\
&\le \sum_{j \in L_i} \delta_j - \sum_{j \in L_i} \rho_{G^{j-1} \cup (S_2^j \backslash A^j)}(A^j) - \rho_S(G') \\
&\le \sum_{j \in L_i} \delta_j - \sum_{j \in L_i} \rho_{N \backslash X_j}(A^j) - \rho_S(G') \\
&\le \sum_{j \in L_i} \delta_j - \sum_{j \in L_i} \rho_{N \backslash X_j}(A^j) - \sum_{j \in L_i} \rho_{N \backslash (X_j \backslash A^j)}(X_j \backslash A^j) \\
&= \sum_{j \in L_i} [\delta_j - \rho_{N \backslash X_j}(X_j)] \\
&= \sum_{j \in L_i} \lambda_j. \qquad (9.4)
\end{aligned}$$

Substituting 9.2, 9.3 and 9.4 into 9.1 yields the statement of the proposition. ■

9.3 Selection of L_i

Different choices of the set L_i could lead to different bounds in Proposition 9.4. The problem of choosing the values of L_i yielding the tightest bound is, in general, nontrivial to solve. For each $i \in \{1, ..., t-1\}$, we can formalize the best choice of L_i by the following mixed-integer program:

$$\min \overline{\gamma}_{i+1}\, d^i_{L_i} + \sum_{j=1}^{i} \lambda_j\, y_j,$$

$$\begin{aligned}
iy &\leq \sum_{j=1}^{i} y_j, \\
d^i_{L_i} &\geq m_{i+1}\,(1-y), \\
d^i_{L_i} &\geq |G^i| - \sum_{j=1}^{i} n_j\, y_j - M\, y, \\
y &\in \{0,1\}, \\
y_j &\in \{0,1\},\ j=1,...,i, \\
d^i_{L_i} &\geq 0.
\end{aligned}$$

where M is a sufficiently large positive scalar. The above optimization problem has $i+1$ binary variables, y and y_j $j = 1, ..., i$, and one continuous variable $d^i_{L_i}$. The condition $y = 1$ is equivalent to the condition $\sum_{j \in L_i} n_j = |G^i|$ is true. The condition $y_j = 1$ represents $j \in L_i$. If $y = 1$ then $y_j = 1$ for all j and the minimization sets $d^i_{L_i} = 0$. Otherwise, the optimization is free to construct L_i and sets $d^i_{L_i} = \max(m_{i+1}, |G^i| - \sum_{j \in L_i} n_j)$ as required.

An alternative to solving the optimization problem above is the following heuristic:

Algorithm 9.1

1. *Order the elements* $\{1, 2, ..., i\}$ *in nondecreasing order of* $\overline{\lambda}_i$. *Let this permutation be denoted* $(p_1, p_2, ..., p_i)$;

2. *For each $j \in \{1, ..., i\}$, compute the bound of Proposition 9.4 with $L_i = \bigcup_{l=1}^{j} p_l$. Call this bound $B(j)$. Define $B(0)$ to equal the bound with $L_i = \phi$;*

3. *Set $L_i := \bigcup_{l=1}^{j^*} p_l$ where $j^* := \arg\min_j B(j)$.*

Note that this procedure yields the best value of L_i when $m_i = 1\ \forall i$. Let the value of L_i computed using the above procedure be denoted $\widetilde{L}_i$.

9.4 Application to Standard Greedy

We next apply Proposition 9.4 to a number of special cases, showing that our result implies the bounds in Table (7.2). We assume throughout that Algorithm 9.1 is used to compute $\widetilde{L}_i$.

First consider the case of STANDARD GREEDY applied to $Z_S|\mathcal{F}_U$. Here $t = K$ and $n_i = m_i = 1$ for all i. Note that in this case $\overline{\gamma}'_i = \overline{\delta}_i$. Applying Proposition 9.4 yields:

$$\begin{aligned} z^* &\leq \min_{i\in\{0,\ldots,t-1\}} [z(G^i) + (K - |G^i|)\,\overline{\delta}_{i+1} + d^i_{\widetilde{L}_i}\,\overline{\gamma}_{i+1} + \sum_{j\in\widetilde{L}_i} \lambda_j] \\ &\leq \min_{i\in\{0,\ldots,t-1\}} [z(G^i) + (K - |G^i|)\,\overline{\delta}_{i+1} + d^i_{\phi}\,\overline{\gamma}_{i+1} + \sum_{j\in\phi} \lambda_j] \\ &= \min_{i\in\{0,\ldots,t-1\}} [z(G^i) + (K - |G^i|)\,\overline{\delta}_{i+1} + |G^i|\,\overline{\gamma}_{i+1}] \\ &= \min_{i\in\{0,\ldots,K-1\}} [z(G^i) + K\,\delta_{i+1}]. \end{aligned}$$

where the second inequality follows from the fact that Algorithm 9.1 is optimal when $m_i = 1$ for all i. The resulting set of inequalities, $z^* \leq \min_{i\in\{0,\ldots,K-1\}} [z(G^i) + K\,\delta_{i+1}]$ leads to a $[1 - [\frac{K-1}{K}]^K]^{-1} < \frac{e}{e-1}$ performance guarantee (see [Wol82] for proof of this statement).

We now consider the case of STANDARD GREEDY applied to $Z_S|\mathcal{F}_M$. Applying Proposition 9.4:

$$\begin{aligned}
z^* &\le \min_{i\in\{0,\dots,t-1\}}[z(G^i)+(K-|G^i|)\,\overline{\delta}_{i+1}+d^i_{\widetilde{L}_i}\,\overline{\gamma}_{i+1}+\sum_{j\in\widetilde{L}_i}\lambda_j] \\
&\le z(G^{t-1})+(K-|G^{t-1}|)\,\overline{\delta}_t+d^{t-1}_{\widetilde{L}_{t-1}}\,\overline{\gamma}_t+\sum_{j\in\widetilde{L}_{t-1}}\lambda_j \\
&\le z(G^{t-1})+(K-|G^{t-1}|)\,\overline{\delta}_t+d^{t-1}_{\{1,\dots,t-1\}}\,\overline{\gamma}_t+\sum_{j=1}^{t-1}\lambda_j \\
&= z(G^{t-1})+(K-|G^{t-1}|)\,\delta_t+\sum_{j=1}^{t-1}\lambda_j \\
&\le z(G^t)+\sum_{j=1}^{t}\lambda_j \\
&\le z(G^t)[1+\alpha].
\end{aligned}$$

where the next to last inequality follows from the fact $\lambda_t \ge 0$. The last inequality follows from the definition of α. Specifically, letting $G^t=\{g_1,g_2,\dots,g_t\}$,

$$\begin{aligned}
\sum_{j=1}^{t}\lambda_j &= \sum_{j=1}^{t}[\delta_j-\rho_{N\backslash X_j}(X_j)], \\
&= z(G^t)-\sum_{j=1}^{t}\rho_{N\backslash g_j}(g_j), \\
&\le z(G^t)-\sum_{j=1}^{t}\rho_{N\backslash g_j}(g_j)\frac{\delta_j}{\rho_\phi(g_j)}, \\
&\le z(G^t)-(1-\alpha)\sum_{j=1}^{t}\delta_j, \\
&= \alpha\, z(G^t).
\end{aligned}$$

Finally, consider the complete enumeration case; that is the case where $t=1$ and $n_1'=K$. Substituting in Proposition 9.4 for $i=1$ and noting that $\gamma_1'=\delta_1$, $\widetilde{L}_0 := \phi$, and $d^i_{\widetilde{L}_i}=0$ yields $z^* \le z^G$ as expected.

Proposition 9.4 provides the most general result of the performance of greedy-type algorithms to Problem (P) when I is a matroid, unifying the results in Table (7.2), and extending them to multi-unit augmentations.

Chapter 10

Extensions

In this chapter, we extend the analysis of the previous chapter in two directions. The first is by defining a continuous extension of Problem (P) and the second is by considering general independence systems. The bounds developed are specific to STANDARD GREEDY.

10.1 Continuous Extension

In this section we derive a specialized version of Proposition 9.4 that bounds the optimal value of a relaxation of Problem (P) with respect to the solution produced by STANDARD GREEDY. This implies that some of the matroid bounds in Table (7.2) hold with respect to the relaxed problem.

We review the following notation: $\mathcal{B} := \{0, 1\}$; M^t denotes the transpose of matrix M; χ_S denotes the incidence vector of $S \subseteq N$; $n := |N|$; $\mathbf{v}_i$ denotes the component of a vector $\mathbf{v} \in \mathcal{R}^n$ corresponding to $i \in N$; $\mathbf{v}(S) := \sum_{i \in S} \mathbf{v}_i$, $\forall S \subseteq N$; and $\mathbf{x} \vee \mathbf{y}$ denotes the vector whose jth component is $\max(\mathbf{x}_j, \mathbf{y}_j)$.

The independent set polytope, P_r, associated with a matroid of rank function r, is the convex hull of the incidence vectors of all independent sets of the matroid. It is well known that the independent set polytope can be expressed as $P_r := \{\mathbf{x} \in R^n_+ : \mathbf{x}(S) \le r(S),\ \forall S \subseteq N\}$. Therefore, the set of incidence vectors of the independent sets of the matroid is given by $P_r \cap \mathcal{B}^n$. Problem (P) can be cast as the following integer program:

$$z^* := \max\{z(S) : \chi_S \in P_r \cap \mathcal{B}^n\}. \qquad (\widetilde{P})$$

Let U denote the $n \times |2^N|$ zero-one matrix whose columns are the incidence vectors of all subsets of N. Similarly, let $\boldsymbol{\zeta} \in \mathcal{R}_+^{|2^N|}$ denote the vector whose elements correspond to the values of the function z, that is $\boldsymbol{\zeta}_S := z(S),\ \forall S \subseteq N$. For each $\mathbf{x} \in [0,1]^n$, define $\Lambda(\mathbf{x}) := \{\boldsymbol{\lambda} \in \mathcal{R}_+^{|2^N|} : U\boldsymbol{\lambda} = \mathbf{x}, \mathbf{1}^T\boldsymbol{\lambda} = 1\}$. We now define the following extension of z. The function $\overline{z} : [0,1]^n \to \mathcal{R}$ is defined as

$$\overline{z}(\mathbf{x}) := \max\{\boldsymbol{\zeta}^T\boldsymbol{\lambda} : \boldsymbol{\lambda} \in \Lambda(\mathbf{x})\}. \qquad (LP1)$$

Clearly, $\overline{z}(\chi_S) = z(S)$ for all $S \subseteq N$. It is worth noting here that for submodular functions, the well-known Lovasz extension is given by $\min\{\boldsymbol{\zeta}^T\boldsymbol{\lambda} : \boldsymbol{\lambda} \in \Lambda(\mathbf{x})\}$.

The following optimization problem is a continuous relaxation of $\widetilde{P}$:

$$\overline{z}^* := \max\{\overline{z}(\mathbf{x}) : \mathbf{x} \in P_r\}. \qquad (\overline{P})$$

Note that $\overline{z}^* \geq z^*$. We next provide an example where $\overline{z}^* > z^*$.

Example 10.1

$N = \{e_1, e_2, e_3, e_4\}$. The function z is defined in the table below.

A	$z(A)$	A	$z(A)$
ϕ	0	$\{e_2, e_3\}$	4
$\{e_1\}$	2	$\{e_2, e_4\}$	4
$\{e_2\}$	2	$\{e_3, e_4\}$	4
$\{e_3\}$	2	$\{e_1, e_2, e_3\}$	6
$\{e_4\}$	3	$\{e_1, e_2, e_4\}$	5
$\{e_1, e_2\}$	4	$\{e_1, e_3, e_4\}$	5
$\{e_1, e_3\}$	4	$\{e_2, e_3, e_4\}$	5
$\{e_1, e_4\}$	4	N	6

It can be verified that this function is normalized, nondecreasing, and submodular. Now define the matroid $I := (N, \mathcal{F})$ by all subsets of N having cardinality 2 or less; that is

$$\mathcal{F} = \{\phi, \{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_1, e_2\}, \{e_1, e_3\}, \{e_1, e_4\}, \{e_2, e_3\}, \{e_2, e_4\}, \{e_3, e_4\}\}.$$

Clearly, $z^* = 4$. However, the optimal solution to Problem $(\overline{P})$ is $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, yielding

$$\begin{aligned} \overline{z}^* &= \overline{z}((\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})) \\ &= \frac{1}{2} z(\{e_1, e_2, e_3\}) + \frac{1}{2} z(\{e_4\}) \\ &= 4.5 \end{aligned}$$

In general, we are not aware of an efficient procedure for computing $\overline{z}^*$. By 'efficient procedure' we mean one that requires a number of calls to the function z that is polynomial in n. Hence our motivation for studying the relaxed problem, $(\overline{P})$, is not to provide upper bounds on the discrete problem, (P), but rather to view it as a problem in its own right arising in certain contexts (such as the example provided next) and to prove that STANDARD GREEDY provides a constant factor approximation.

Example 10.2

Bikhchandani and Mamer [BiM97] address the problem of existence of competitive equilibria in exchange economies. For purposes of this example, we restrict attention to the case of a single seller, multiple buyers, and one unit per item. This is essentially the set up of the combinatorial auction problem discussed in Section 6.3.1. We have seen that finding an efficient allocation for this problem is a special case of Problem (P). In [BiM97], the existence of market clearing prices is related to the optimal value of an accompanying problem where items are made 'divisible'; that is, agents can be assigned fractions of bundles. It can be shown that, in this context, the allocation problem with indivisibilities corresponds exactly to Problem $(\overline{P})$. Bikhchandani and Mamer [BiM97] show that market clearing prices exist if and only if $z^* = \overline{z}^*$; that is indivisibilities do not yield any improvement in allocation efficiency. Kelso and Crawford [KeC82] provide an example with submodular preferences where $z^* < \overline{z}^*$. Lehmann [Leh02] shows that, for submodular preferences, the *allocation value, z^G, produced by the sequential single single-item auction mechanism outlined in Section 6.3.1 is bounded as follows: $z^G \geq \frac{1}{2}\overline{z}^*$. This strengthens Proposition 6.4 by Lehmann, Lehmann, and Nisan [LLN02] and implies $z^* > \frac{1}{2}\overline{z}^*$. This result is generalized in Proposition 10.8 of this section.*

The next two propositions show that the extension function $\overline{z}$ inherits from the original function z the monotonicity property as well as a form of 'continuous submodularity'.

Proposition 10.1

If $\mathbf{0} \leq \mathbf{x} \leq \mathbf{y} \leq \mathbf{1}$ then $\overline{z}(\mathbf{x}) \leq \overline{z}(\mathbf{y})$.

Proof.

First consider the case where $\mathbf{y} = \mathbf{x} + \theta\,\mathbf{e}$ where $\theta \geq 0$ and $\mathbf{e}$ is a unit vector in the *ith* direction. Suppose $\overline{z}(\mathbf{x}) = \boldsymbol{\zeta}^T \boldsymbol{\lambda}$ where $\boldsymbol{\lambda} \in \Lambda(\mathbf{x})$. Unless $\mathbf{x} = \mathbf{1}$, there must exist components $\boldsymbol{\lambda}_i$ of the vector $\boldsymbol{\lambda}$ whose *ith* component is equal to zero. By switching the value of some of these components to 1 (and possibly splitting some of the $\boldsymbol{\lambda}_i$) one can construct a new vector $\boldsymbol{\lambda}' \in \Lambda(\mathbf{y})$. By monotonicity of z, $\overline{z}(\mathbf{x}) \leq \boldsymbol{\zeta}^T \boldsymbol{\lambda}' \leq \overline{z}(\mathbf{y})$. Finally, we note that for any $\mathbf{x} \leq \mathbf{y}$, repeating the above argument component-wise proves the proposition. ■

The following result is an intermediate step in proving Proposition 10.3.

Proposition 10.2

Let $\mathbf{0} \leq \mathbf{x} \leq \mathbf{y} \leq \mathbf{y} + \boldsymbol{\delta} \leq \mathbf{1}$ where $\mathbf{x} \in \mathcal{B}^n$. Suppose that $\overline{z}(\mathbf{y} + \boldsymbol{\delta}) = \boldsymbol{\zeta}^T \boldsymbol{\lambda}$ where $\boldsymbol{\lambda} \in \Lambda(\mathbf{y} + \boldsymbol{\delta})\}$. Then there exists scalars $\pi_{\mathbf{u},\mathbf{v}}$, $\forall \mathbf{u}, \mathbf{v} \in \mathcal{B}^n$ where $\mathbf{x} \leq \mathbf{v} \leq \mathbf{u}$, satisfying:

$$\sum_{\mathbf{u} \in B^n} \sum_{\mathbf{v} \in B^n : \mathbf{x} \leq \mathbf{v} \leq \mathbf{u}} \pi_{\mathbf{u},\mathbf{v}}\, \mathbf{v} = \mathbf{y},$$

$$\sum_{\mathbf{v} \in B^n : \mathbf{x} \leq \mathbf{v} \leq \mathbf{u}} \pi_{\mathbf{u},\mathbf{v}} = \boldsymbol{\lambda}_S,\ \forall S \subseteq N, \mathbf{u} = \boldsymbol{\chi}_S,$$

$$\pi_{\mathbf{u},\mathbf{v}} \geq 0,\ \forall \mathbf{u}, \mathbf{v} \in \mathcal{B}^n, \mathbf{x} \leq \mathbf{v} \leq \mathbf{u}.$$

Proof.

First note that $\chi_S \geq \mathbf{x}$ for all $S \subseteq N$ such that $\boldsymbol{\lambda}_S > 0$ because $\mathbf{1}^T\boldsymbol{\lambda} = 1$. We next prove the proposition by induction on the number, k, of non-zero components of the vector $\boldsymbol{\delta}$. The proposition obviously holds for $k = 0$ because, in this case, $\boldsymbol{\delta} = \mathbf{0}$ and setting $\pi_{\mathbf{u},\mathbf{v}} = \boldsymbol{\lambda}_S$ if $\mathbf{v} = \mathbf{u} = \chi_S$, and 0 otherwise, satisfies the statement of the proposition. Now suppose the proposition holds for $k = k'$. For $k = k' + 1$, assume without loss of generality that the first $k' + 1$ elements of $\boldsymbol{\delta}$ are positive. Let δ_1 be the value of the first component of $\boldsymbol{\delta}$, let $\boldsymbol{\delta}'$ be the vector derived from $\boldsymbol{\delta}$ by replacing δ_1 by 0, and set $\mathbf{y}' := \mathbf{y} + (\boldsymbol{\delta} - \boldsymbol{\delta}')$. Applying the induction hypothesis to $\mathbf{0} \leq \mathbf{x} \leq \mathbf{y}' \leq \mathbf{y} + \boldsymbol{\delta} = \mathbf{y}' + \boldsymbol{\delta}' \leq \mathbf{1}$, there exists scalars $\pi'_{\mathbf{u},\mathbf{v}}$, $\forall \mathbf{u}, \mathbf{v} \in B^n$ where $\mathbf{x} \leq \mathbf{v} \leq \mathbf{u}$, satisfying:

$$\sum_{\mathbf{u} \in B^n} \sum_{\mathbf{v} \in B^n : \mathbf{x} \leq \mathbf{v} \leq \mathbf{u}} \pi'_{\mathbf{u},\mathbf{v}} \mathbf{v} = \mathbf{y}',$$

$$\sum_{\mathbf{v} \in B^n : \mathbf{x} \leq \mathbf{v} \leq \mathbf{u}} \pi'_{\mathbf{u},\mathbf{v}} = \boldsymbol{\lambda}_S, \ \forall S \subseteq N, \mathbf{u} = \chi_S,$$

$$\pi'_{\mathbf{u},\mathbf{v}} \geq 0, \ \forall \mathbf{u}, \mathbf{v} \in B^n, \mathbf{x} \leq \mathbf{v} \leq \mathbf{u}.$$

$\mathbf{y}'$ differs from $\mathbf{y}$ in only the first component. Since $\delta_1 > 0$, there exists a pair $\mathbf{u}, \mathbf{v}$ such that the first component of $\mathbf{v}$ equals 1 and $\pi'_{\mathbf{u},\mathbf{v}} > 0$. Let $\mathbf{v}'$ be the vector derived from $\mathbf{v}$ by changing the first component from 1 to 0. Perform the following update: $\pi'_{\mathbf{u},\mathbf{v}} := \pi'_{\mathbf{u},\mathbf{v}} - \min(\pi'_{\mathbf{u},\mathbf{v}}, \delta_1)$ and $\pi'_{\mathbf{u},\mathbf{v}'} := \pi'_{\mathbf{u},\mathbf{v}'} + \min(\pi'_{\mathbf{u},\mathbf{v}}, \delta_1)$. By repeating this process a finite number of times, we end up with a set of scalars that satisfy the conditions of the proposition. ■

Proposition 10.3

Suppose $\mathbf{0} \leq \mathbf{x} \leq \mathbf{y} \leq \mathbf{y} + \boldsymbol{\delta} \leq \mathbf{1}$ *where* $\mathbf{x} \in B^n$. *Then the following inequality holds:*

$$\overline{z}(\mathbf{y} + \boldsymbol{\delta}) - \overline{z}(\mathbf{y}) \leq \overline{z}(\mathbf{x} + \boldsymbol{\delta}) - \overline{z}(\mathbf{x}).$$

Proof.

Let $\overline{z}(\mathbf{y} + \boldsymbol{\delta}) = \boldsymbol{\zeta}^T\boldsymbol{\lambda}$ where $\boldsymbol{\lambda} \in \Lambda(\mathbf{y} + \boldsymbol{\delta})\}$. Let $\pi_{\mathbf{u},\mathbf{v}}$ be a set of scalars satisfying the conditions of Proposition 10.2. Therefore,

$$\begin{aligned}
\overline{z}(\mathbf{y}+\boldsymbol{\delta}) - \overline{z}(\mathbf{y}) &= \sum_{S\subseteq N} \boldsymbol{\lambda}_S\, z(S) - \overline{z}(\mathbf{y}) \\
&= \sum_{\mathbf{u}\in B^n} \sum_{\mathbf{v}\in B^n:\mathbf{x}\leq\mathbf{v}\leq\mathbf{u}} \pi_{\mathbf{u},\mathbf{v}}\, \widetilde{z}(\mathbf{u}) - \overline{z}(\mathbf{y}) \\
&\leq \sum_{\mathbf{u}\in B^n} \sum_{\mathbf{v}\in B^n:\mathbf{x}\leq\mathbf{v}\leq\mathbf{u}} \pi_{\mathbf{u},\mathbf{v}}\, \widetilde{z}(\mathbf{u}) - \sum_{\mathbf{u}\in B^n} \sum_{\mathbf{v}\in B^n:\mathbf{x}\leq\mathbf{v}\leq\mathbf{u}} \pi_{\mathbf{u},\mathbf{v}}\, \widetilde{z}(\mathbf{v}) \\
&= \sum_{\mathbf{u}\in B^n} \sum_{\mathbf{v}\in B^n:\mathbf{x}\leq\mathbf{v}\leq\mathbf{u}} \pi_{\mathbf{u},\mathbf{v}} [\widetilde{z}(\mathbf{u}) - \widetilde{z}(\mathbf{v})] \\
&\leq \sum_{\mathbf{u}\in B^n} \sum_{\mathbf{v}\in B^n:\mathbf{x}\leq\mathbf{v}\leq\mathbf{u}} \pi_{\mathbf{u},\mathbf{v}} [\widetilde{z}(\mathbf{x}+(\mathbf{u}-\mathbf{v})) - \widetilde{z}(\mathbf{x})]
\end{aligned}$$

where the last inequality follows from the submodularity of z, the definition of $\widetilde{z}$, and the fact that $\mathbf{x} \leq \mathbf{v} \leq \mathbf{u}$. Note that the scalars $\pi_{\mathbf{u},\mathbf{v}}$ form a set of valid convexity weights because they are nonnegative and $\sum_{\mathbf{u}\in B^n} \sum_{\mathbf{v}\in B^n:\mathbf{x}\leq\mathbf{v}\leq\mathbf{u}} \pi_{\mathbf{u},\mathbf{v}} = \sum_{S\subseteq N} \boldsymbol{\lambda}_S = 1$. Also, note that

$$\sum_{\mathbf{u}\in B^n} \sum_{\mathbf{v}\in B^n:\mathbf{x}\leq\mathbf{v}\leq\mathbf{u}} \pi_{\mathbf{u},\mathbf{v}}(\mathbf{x}+\mathbf{u}-\mathbf{v}) = \mathbf{x} + (\mathbf{y}+\boldsymbol{\delta}) - \mathbf{y} = \mathbf{x}+\boldsymbol{\delta}.$$

Therefore,

$$\sum_{\mathbf{u}\in B^n} \sum_{\mathbf{v}\in B^n:\mathbf{x}\leq\mathbf{v}\leq\mathbf{u}} \pi_{\mathbf{u},\mathbf{v}} \widetilde{z}(\mathbf{x}+(\mathbf{u}-\mathbf{v})) \leq \overline{z}(\mathbf{x}+\boldsymbol{\delta}).$$

Combining this with the above inequality yields the statement of the proposition. ∎

We conjecture the following generalization of Proposition 10.3:

Conjecture 10.1

Suppose $\mathbf{0} \leq \mathbf{x} \leq \mathbf{y} \leq \mathbf{y}+\boldsymbol{\delta} \leq \mathbf{1}$. *Then the following inequality holds:*

$$\overline{z}(\mathbf{y}+\boldsymbol{\delta}) - \overline{z}(\mathbf{y}) \leq \overline{z}(\mathbf{x}+\boldsymbol{\delta}) - \overline{z}(\mathbf{x}).$$

A corollary of Proposition 10.3 is that $\overline{z}(\cdot)$ is subadditive.

Proposition 10.4

Suppose $\boldsymbol{\theta} = \sum_{i=1}^{t} \boldsymbol{\theta}^i \leq \mathbf{1}$ *for some* $\boldsymbol{\theta}^i \in \mathcal{R}^n_+$, $i = 1, \ldots t$. *Then,* $\overline{z}(\boldsymbol{\theta}) \leq \sum_{i=1}^{t} \overline{z}(\boldsymbol{\theta}^i)$.

Proof.

Consider first the case where $t = 2$. Substitute $\mathbf{x} = \mathbf{0}$, $\mathbf{y} = \boldsymbol{\theta}^1$, and $\boldsymbol{\delta} = \boldsymbol{\theta}^2$ in Proposition 10.3. Because $\overline{z}(\mathbf{0}) = 0$, this yields $\overline{z}(\boldsymbol{\theta}) \leq \overline{z}(\boldsymbol{\theta}^1) + \overline{z}(\boldsymbol{\theta}^1)$. The result follows by induction on t. ■

We next state a well known result pertaining to polymatroids. For a proof, the reader is referred to [Sch03].

Proposition 10.5 *(see [Sch03])*

Let P_r be a polymatroid associated with some matroid whose rank function is $r : 2^N \rightarrow \mathcal{R}$. An element $\mathbf{x} \in P_r$ is maximal (with respect to $\leq$) if and only if $x(N) = r(N)$.

The following *exchange* result is due to Faigle [Fai84] and provides a continuous analogue to Proposition 9.1:

Proposition 10.6 *(Faigle [Fai84])*

Let P_r be any polymatroid. If $\mathbf{x}, \mathbf{y} \in P_r$ with $\mathbf{x} = \mathbf{x}^1 + \mathbf{x}^2$ where $\mathbf{x}^1, \mathbf{x}^2 \in P_r$, then there exists $\mathbf{y}^1, \mathbf{y}^2 \in P_r$ such that $\mathbf{y} = \mathbf{y}^1 + \mathbf{y}^2$, and $\mathbf{x}^1 + \mathbf{y}^1$, $\mathbf{x}^2 + \mathbf{y}^2 \in P_r$.

A simple approach to finding a good feasible solution to Problem $(\widetilde{P})$ (equivalently, Problem (P)) is the standard best-in greedy algorithm expressed below in terms vector notation:

Algorithm 10.1 Vector Standard Greedy

Initialize $i := 0$; *Select* $\mathbf{g}^0 := \mathbf{0}$;

While $\mathbf{g}^i(N) < r(N)$ *do*

{ *Set* $i := i + 1$;

Select $\mathbf{x}^i = \arg\max \{\overline{z}(\mathbf{g}^{i-1} + \boldsymbol{\delta}) - \overline{z}(\mathbf{g}^{i-1}) : \mathbf{g}^{i-1} + \boldsymbol{\delta} \in P_r$ *and* $\boldsymbol{\delta}(N) = 1\}$;

Set $\mathbf{g}^i := \mathbf{g}^{i-1} + \mathbf{x}^i$. }.

It is important to note that at each iteration of the above algorithm, an integral augmentation $\mathbf{x}^i$ exists that maximizes marginal improvement. Therefore, $\mathbf{g}^i$ is integral at each stage of the algorithm and the sequence of partial solutions produced corresponds exactly to that generated by STANDARD GREEDY when applied to Problem (P).

We now state the two main results of this section.

Proposition 10.7

Suppose VECTOR STANDARD GREEDY *is applied to solve Problem $(\widetilde{P})$ assuming the underlying matroid is uniform. Let z^G denote the value of the solution produced. Then*

$$\frac{\overline{z}^*}{z^G} \le \frac{e}{e-1}.$$

Proof.

Let $t := r(N)$. Because the function $\overline{z}$ is nondecreasing (Proposition 10.1), we can restrict attention to a maximal optimal solution, $\mathbf{w}$, where $\mathbf{w}(N) = t$ by Proposition 10.5. Therefore, $\mathbf{w}$ can be expressed as $\sum_{i=1}^{t} \mathbf{w}^i$ where $\mathbf{w}^i(N) = 1\ \forall i$. Let $\mathbf{g}$ denote the solution produced by VECTOR STANDARD GREEDY. Clearly, this solution is also maximal. Furthermore, $\mathbf{g}$ is integral.

For each $i \in \{0, ..., t-1\}$ the following relations hold:

$$\begin{aligned}
\overline{z}(\mathbf{w}) &= \overline{z}(\mathbf{g}^i) + \rho_{\mathbf{g}^i}(\mathbf{w}) \\
&\le \overline{z}(\mathbf{g}^i) + \textstyle\sum_{j=1}^{t} \rho_{\mathbf{g}^i}(\mathbf{w}^j) \\
&\le \overline{z}(\mathbf{g}^i) + \textstyle\sum_{j=1}^{t} \rho_{\mathbf{g}^i}(\mathbf{x}^i) \\
&= \overline{z}(\mathbf{g}^i) + t\, \rho_{\mathbf{g}^i}(\mathbf{x}^i).
\end{aligned}$$

The first inequality holds by Proposition 10.1, the second inequality holds by Proposition 10.4, and the final inequality holds by definition of VECTOR STANDARD GREEDY. The t inequalities formed above by taking $i = 0, 1, ..., t-1$, was shown by Wolsey [Wol82] to imply a bound of $e/(e-1)$ on $\overline{z}(\mathbf{w})/\overline{z}(\mathbf{g})$. ∎

Proposition 10.8

Suppose VECTOR STANDARD GREEDY *is applied to solve Problem $(\widetilde{P})$. Let z^G denote* *the value of the solution produced. Then*

$$\frac{\bar{z}^*}{z^G} \leq 2.$$

Proof.

Let $t := r(N)$. Because the function $\bar{z}$ is nondecreasing (Proposition 10.1), we can restrict attention to a maximal optimal solution, $\mathbf{w}$, where $\mathbf{w}(N) = t$ by Proposition 10.5. Let $\mathbf{g}$ denote the solution produced by GREEDY($\mathbf{m}, \mathbf{n}$) with $\mathbf{n} = \mathbf{1}$. Clearly, this solution is also maximal.

Let $\boldsymbol{\theta} := (\mathbf{w} \vee \mathbf{g}) - \mathbf{g}$. Note that $\boldsymbol{\theta} \leq \mathbf{w}$ and, therefore, $\boldsymbol{\theta}' \in P_r$. We will assume that $\boldsymbol{\theta}(N) = t$. Recall that $\mathbf{g}^i$ denotes the partial greedy solution constructed at the end of iteration i. Applying Proposition 10.6 recursively, there exists a collection of vectors $\{\boldsymbol{\theta}^i\}_{i=1}^t$ with the property that:

$$\boldsymbol{\theta} := \boldsymbol{\theta}^1 + \boldsymbol{\theta}^2 + \cdots + \boldsymbol{\theta}^t; \text{ and}$$
$$\mathbf{g}^{i-1} + \boldsymbol{\theta}^i \in P_r \ \forall i = 1, ..., t.$$

Let $\Theta^i := \sum_{j=1}^{i} \boldsymbol{\theta}^j$. We have,

$$\begin{aligned}
\bar{z}^* &= \bar{z}(\mathbf{w}) \\
&\leq \bar{z}(\mathbf{w} \vee \mathbf{g}) \\
&= \bar{z}(\mathbf{g} + \boldsymbol{\theta}) \\
&= \bar{z}(\mathbf{g}) + [\sum_{i=1}^{t} \bar{z}(\mathbf{g} + \Theta^{\mathbf{i}}) - \bar{z}(\mathbf{g} + \Theta^{\mathbf{i-1}})] \\
&\leq \bar{z}(\mathbf{g}) + [\sum_{i=1}^{t} \bar{z}(\mathbf{g}^{i-1} + \boldsymbol{\theta}^{\mathbf{i}}) - \bar{z}(\mathbf{g}^{i-1})] \\
&\leq \bar{z}(\mathbf{g}) + [\sum_{i=1}^{t} \bar{z}(\mathbf{g}^{i-1} + \mathbf{x}^{\mathbf{i}}) - \bar{z}(\mathbf{g}^{i-1})] \\
&= 2\,\bar{z}(\mathbf{g}).
\end{aligned}$$

The next to last inequality follows from Proposition 10.3 by taking $\mathbf{x} := \mathbf{g}^{i-1}$, $\mathbf{y} := \mathbf{g} + \Theta^{\mathbf{i-1}}$, and $\boldsymbol{\delta} := \boldsymbol{\theta}^{\mathbf{i}}$. The last inequality follows from the greedy property of the algorithm. ∎

10.2 Independence Systems

In this section, we consider general independence systems and construct a bound on the performance of STANDARD GREEDY based on the rank quotient q. The proof of this result builds on ideas used by Hausmann, Korte, and Jenkyns [HKJ80] and Conforti and Cornuéjols [CoC84].

Proposition 10.9

Let $I = (N, \mathcal{F})$ be an independence system of rank quotient q and let $z : 2^N \rightarrow \mathcal{R}$ be a normalized nondecreasing submodular function. Suppose STANDARD GREEDY *is applied to Problem (P). Then, the following bound holds:*

$$\frac{z^*}{z^G} \leq \frac{1}{q} + \alpha.$$

Proof.

Let $G = \{g_1, g_2, ..., g_t\}$ denote the maximal solution produced by STANDARD GREEDY, and for any $i \in \{1, ..., t\}$ let $G^i = \{g_1, g_2, ..., g_i\}$ denote the partial solution at the end of iteration i. $G^0 = \phi$. Let S denote a maximal optimal solution. For each $i \in \{0, ..., t\}$, define $S^i \subseteq S \backslash G^i$ to be the maximal subset of $S \backslash G^i$ for which $G^i \cup \{e\} \in \mathcal{F}$ $\forall e \in S^i$. By the independence assumption,

$$S = S^0 \supseteq S^1 \supseteq \cdots \supseteq S^t = \phi.$$

Let $\{s^i\}_{i=1}^t$ be the partition of S defined by $s^i := S^{i-1}\backslash S^i$. Note that if $g_i \in S$ then $g_i \in s^i$. A bound on z^* can be derived as follows:

$$\begin{aligned}
z^* &\equiv z(S) \\
&= z(S \cup G) - \rho_S(G\backslash S) \\
&= z(G) + \rho_G(S\backslash G) - \rho_S(G\backslash S) \\
&\leq z(G) + \sum_{i=1}^t \rho_G(s^i\backslash g_i) - \rho_S(G\backslash S) \\
&\leq z(G) + \sum_{i=1}^t \rho_{G^{i-1}}(s^i\backslash g_i) - \rho_S(G\backslash S) \\
&= z(G) + \sum_{i=1}^t [\rho_{G^{i-1}}(s^i) - \rho_{G^{i-1}\cup(s^i\backslash g_i)}(s^i \cap g_i)] - \rho_S(G\backslash S) \\
&\leq z(G) + \sum_{i=1}^t \rho_{G^{i-1}}(s^i) - \sum_{i=1}^t \rho_{N\backslash g_i}(s^i \cap g_i) - \sum_{g_i \notin S} \rho_{N\backslash g_i}(g_i) \\
&\leq z(G) + \sum_{i=1}^t \rho_{G^{i-1}}(s^i) - \sum_{i=1}^t \rho_{N\backslash g_i}(g_i) \\
&\leq z(G) + \sum_{i=1}^t |s^i|\, \delta_i - \sum_{i=1}^t \rho_{N\backslash g_i}(g_i). \\
&= \sum_{i=1}^t |s^i|\, \delta_i + \sum_{i=1}^t \lambda_i.
\end{aligned}$$

where the last inequality follows from the fact that $G^{i-1} \cup s^{i-1} \in \mathcal{F}$.

By definition of the total curvature α, $\sum_{i=1}^t \lambda_i \leq \alpha\, z^G$ (see Section 9.4 for details). To bound the term $\sum\limits_{i=1}^t |s^i|\, \delta_i$, we note that for each $i \in \{1, ..., t\}$, G^i is maximal in $G^i \cup (S\backslash S^i)$ by definition of S^i. Furthermore, by independence, $S\backslash S^i \in \mathcal{F}$. Therefore, $|S\backslash S^i| \leq i/q$. By Proposition 8.1, the worst bound would correspond to $|s^i| = 1/q$ for all i. To summarize,

$$\begin{aligned}
z^* &\leq \sum_{i=1}^t |s^i|\, \delta_i + \sum_{i=1}^t \lambda_i \\
&\leq \frac{1}{q} z^G + \alpha\, z^G,
\end{aligned}$$

yielding the result of the Proposition. ■

Recalling the result by Hausmann at al. stated in Proposition 6.3 that $q \geq \frac{1}{p}$, where p is the minimum number of matroids into which I can be decomposed, the bound of

Proposition 10.9 is stronger than the $p + \alpha$ bound of Conforti and Cornuéjols [CoC84]. It should be pointed out that for some independence systems the difference between $\frac{1}{q}$ and p can be arbitrarily large. Korte and Hausmann [KoH78] have shown that $\frac{1}{q} \leq 2$ for the matching independence system. More recently, Fekete et al. [FFS03] have shown that $p \in \Omega(\log \log n)$ for matching independence systems where n is the number of vertices. Therefore, for large graphs, the difference between both bounds is arbitrarily large.[1]

[1] I thank Professor Andreas Schulz for pointing out the paper by Fekete et al. and its implication for the strength of this bound.

Chapter 11

Part II Summary and Future Directions

11.1 Summary of Contributions

The main contributions of Part II research can be summarized as follows:

1. We analyze the performance of a generalized greedy algorithm that incrementally augments the current solution by adding subsets of arbitrary variable cardinality. At one extreme this algorithm is the same as STANDARD GREEDY. At the other extreme it is synonymous with complete enumeration. This yields a family of algorithms that represents a continuum of trade-offs between optimality and tractability. This analysis is restricted to the case where I is a matroid;

2. We define a continuous relaxation of the discrete problem of maximizing a submodular function over a matroid, and show that previously known performance bounds for STANDARD GREEDY when I is a matroid in fact apply with respect to the relaxed problem. This is analogous to bounding the gap between the greedy solution and the LP relaxation for integer programs;

3. We derive a new performance bound for STANDARD GREEDY when I is a general independence system. The bound is in terms of the rank quotient of the independence system and strengthens a previously known result.

11.2 Future Directions

We conclude by posing a number of extensions and open problems emerging from this research:

- How can Proposition 9.4 be extended to general independence systems yielding, as a special case, Proposition 10.9?
- What is a good sequence of constrained and unconstrained step sizes to use GENERALIZED GREEDY? Do some choices dominate others in terms of bound quality and tractability? How can the choice of step sizes be performed dynamically?
- Is there a polynomial time algorithm for Problem (P) that has a constant factor performance guarantee better than 2? Alternatively, is there a hardness of approximation threshold of 2 under the assumption $P \neq NP$?
- Are the bounds on $\overline{z}^*/z^*$implied by the results of Chapter 10 tight?

Bibliography

[BiM97] S. BIKHCHANDANI and J. W. MAMER. 1997. "Competitive Equilibrium in an Exchange Economy with Indivisibilities". *Journal of Economic Theory*, **74**, 385-413.

[Bry73] T. H. BRYLAWSKI. 1973. "Some Properties of Basic Families of Subsets". *Discrete Mathematics*, **6**, 333-341.

[CoC84] M. CONFORTI and G. CORNUÉJOLS. 1984. "Submodular Set Functions, Matroids and the Greedy Algorithm: Tight Worst-Case Bounds and Some Generalizations of the Rado-Edmonds Theorem". *Discrete Applied Mathematics*, **7**, 251-274.

[CFN77] G. CORNUÉJOLS, M. L. FISHER and G. L. NEMHAUSER. 1977. "Location of Bank Accounts to Optimize Floats: An Analytic Study of Exact and Approximate Algorithms". *Management Science*, **23**, 789-810.

[DoK93] G. DOBSON and S. KALISH. 1993. "Heuristics for Pricing and Positioning a Product-Line Using Conjoint and Cost Data". *Management Science*, **39**, 160-175.

[Edm71] J. EDMONDS. 1971. "Matroids and the Greedy Algorithm". *Mathematical Programming*, **1**, 127-136.

[Fai84] U. FAIGLE. 1984. "A Vector Exchange Property of Submodular Systems". *Discrete Applied Mathematics*, **9**, 209-211.

[Fei78] U. FEIGE. 1998. "A Threshold of $ln\ n$ for Approximating Set Cover". *Journal of the ACM*, **45**, 634-652.

[FFS03] S. P. FEKETE, R. T. FIRLA, and B. SPILLE. 2003. "Characterizing Matchings as the Intersection of Matroids". *Mathematical Methods of Operations Research*, **58**, 319-329.

[FNW78] M. L. FISHER, G. L. NEMHAUSER, and L. A. WOLSEY. 1978. "An Analysis of Approximations for Maximizing Submodular Set Functions - II". *Mathematical Programming Study*, **8**, 73-87.

[Fuj91] S. FUJISHIGE. 1991. "Submodular Functions and Optimization". *Annals of Discrete Mathematics*, **47**.

[Gal68] D. GALE. 1968. "Optimal Assignments in an Ordered Set: An Application of Matroid Theory". *Journal of Combinatorial Theory*, **4**, 176-180.

[Gre73] C. GREENE. 1973. "A Multiple Exchange Property for Bases". *Proceedings of the American Mathematical Society*, **39**, 45-50.

[GrM75] C. GREENE and T. L. MAGNANTI. 1975. "Some Abstract Pivot Algorithms". *SIAM Journal on Applied Mathematics*, **29**, 530-539.

[HKJ80] D. HAUSMANN, B. KORTE and T. A. JENKYNS. 1980. "Worst Case Analysis of Greedy Type Algorithms for Independence Systems". *Mathematical Programming Study*, **12**, 120-131.

[Hoc97] D. HOCHBAUM, Ed. 1997. *Approximation Algorithms for NP-Hard Problems.* PWS.

[HoM85] D. S. HOCHBAUM and W. MAASS. 1985. "Approximation Schemes for Covering and Packing Problems in Image Processing and VLSI". *Journal of the ACM*, **32**, 130-136.

[IbK75] O. H. IBARRA and C. E. KIM. 1975. "Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems". *Journal of the ACM*, **22**, 463-468.

[Jen76] T. A. JENKYNS. 1976. "The Efficacy of the 'Greedy' Algorithm". *Proceedings of the Seventh Southeastern Conference on Combinatorics, Graph Theory, and Computing*, 341-350.

[Joh74] D. S. JOHNSON. 1974. "Approximation Algorithms for Combinatorial Problems". *Journal of Computer and System Sciences*, **9**, 256-278.

[KaP03] J. KALAGNANAM and D. PARKES. 2003. "Auctions, Bidding and Exchange Design", Chapter 1 in *Supply Chain Analysis in the eBusiness Era*, D. Simchi-Levi, S. D. Wu, and Z. M. Shen Eds. Forthcoming.

[Kar72] R. M. KARP. 1972. "Reducibility Among Combinatorial Problems" in R. E. MILLER and J. W. THATCHER (eds.), *Complexity of Computer Computations*, 85-103.

[KeC82] A. S. KELSO and V. P. CRAWFORD. 1982. "Job Matching, Coalition Formation, and Gross Substitutes". *Econometrica*, **50**, 1483-1504.

[KMN99] S. KHULLER, A. MOSS and J. NAOR. 1999. "The Budgeted Maximum Coverage Problem". *Information Processing Letters*, **70**, 39-45.

[KPR98] J. KLEINBERG, C. PAPADIMITRIOU and P. RAGHAVAN. 1998. "Segmentation Problems". *Proceedings of STOC 1998*, 473-482.

[Kle99] P. KLEMPERER. 1999. "Auction Theory: A Guide to the Literature". *Journal of Economic Surveys*, **13**, 227-286.

[KoH78] B. KORTE and D. HAUSMANN. 1978. "An Analysis of the Greedy Heuristic for Independence Systems". *Annals of Discrete Mathematics*, **2**, 65-74.

[LLN02] B. LEHMANN, D. LEHMANN and N. NISAN. 2002. "Combinatorial Auctions with Decreasing Marginal Utilities". *Games and Economic Behavior.* Forthcoming.

[Leh02] D. LEHMANN. 2002. "Bounding the Integral Gap for Submodular Economies". *Personal Communication.*

[Lov83] L. LOVASZ. 1983. "Submodular Functions and Convexity". In *Mathematical Programming - The State of the Art*, edited by A. Bachem, M. Grotschel, and B. Korte. Springer.

[Nar97] A. NARAYANAN. 1997. "Submodular Functions and Electrical Networks". *Annals of Discrete Mathematics*, **54**.

[NeW78] G. L. NEMHAUSER and L. A. WOLSEY. 1978. "Best Algorithms for Approximating the Maximum of a Submodular Set Function". *Mathematics of Operations Research*, **3**, 177-188.

[NeW88] G. L. NEMHAUSER and L. A. WOLSEY. 1988. *Integer and Combinatorial Optimization.* Wiley.

[NWF78] G. L. NEMHAUSER, L. A. WOLSEY and M. L. FISHER. 1978. "An Analysis of Approximations for Maximizing Submodular Set Functions - I". *Mathematical Programming*, **14**, 265-294.

[Rad57] R. RADO. 1957. "Note on Independence Functions". *Proceedings of the London Mathematical Society*, **7**, 300-320.

[Sch03] A. SCHRIJVER. 2003. *Combinatorial Optimization: Polyhedra and Efficiency.* Springer.

[Sim89] H. U. SIMON. 1989. "Approximation Algorithms for Channel Assignment in Cellular Radio Networks". *Lecture Notes in Computer Science*, **380**, 405-415.

[Vaz01] V. VAZIRANI. 2001. *Approximation Algorithms.* Springer.

[Wol82] L. WOLSEY. 1982. "Maximizing Real-Valued Submodular Functions: Primal and Dual Heuristics for Location Problems". *Mathematics of Operations Research*, **7**, 410-425.

[Woo74] D. R. WOODALL. 1974. "An Exchange Theorem for Bases of Matroids". *Journal of Combinatorial Theory, Series B*, **16**, 227-228.