Probabilistic On-line Transportation Problems
with Carrying-Capacity Constraints

by

Kyle Treleaven

M.S., University of Pittsburgh (2007)
B.S., University of Pittsburgh (2006)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

© Massachusetts Institute of Technology 2014. All rights reserved.

Signature redacted

Author ........................................
Department of Aeronautics and Astronautics
May 21, 2014

Signature redacted

Certified by ........................................
Emilio Frazzoli
Professor of Aeronautics and Astronautics, MIT
Thesis Supervisor

Signature redacted

Certified by ........................................
Dimitris Bertsimas
Professor of Operations Research and Statistics, MIT

Signature redacted

Certified by ........................................
Patrick Jaillet
Professor of Electrical Engineering and Computer Science, MIT

Signature redacted

Accepted by ........................................
Paul C. Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee
Probabilistic On-line Transportation Problems with Carrying-Capacity Constraints

by

Kyle Treleaven

Submitted to the Department of Aeronautics and Astronautics on May 21, 2014, in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

This thesis presents new insights and techniques for the analysis and design of autonomous or technology-assisted (“intelligent”) transportation systems. The focus is on cooperative, on-line planning and control, of a fleet of transport vehicles with limited carrying capacity, where new transportation demands enter the system in real time. The study extends an existing probabilistic framework which has provided numerous insights about vehicle scheduling and routing problems since its inception. Additionally, the thesis provides algorithms and new probabilistic cost bounds, for optimal bipartite matchings between large sets of random points and optimal stacker crane tours through large sets of random demands.

A recurrent theme of the thesis is that capacity-constrained vehicles must drive passenger-less, inescapably, for some positive fraction of time (in almost any practical setting). Moreover, under probabilistic modelling for the uncertainty of demand, one can predict the aforementioned fraction precisely, using strong Laws of Large Numbers arguments; it relates to a quantity known as the Earth Mover’s distance (EMD), described by a fundamental problem in transportation theory. Since the existence of an unavoidable extra cost term has significant implications, e.g., for operational budgets of shared-vehicle systems, the results illuminate a phenomenon whose neglect could prove an unfortunate oversight. To the author’s knowledge, this connection of the EMD to on-line vehicle routing is novel.

The thesis also provides a new study of the practical considerations imposed by the “street rules” ubiquitous among ground-based transport problems. A new efficient algorithm for the Bipartite Matching problem for points on a roadmap is given. Also given is a new explicit formulation of the EMD on road networks; very few explicit formulas for EMDs have been known previously.

Thesis Supervisor: Emilio Frazzoli
Title: Professor of Aeronautics and Astronautics, MIT
Acknowledgements

There is no doubt I must express, first, my deep appreciation to my advisor, Emilio Frazzoli. Without Emilio’s unflagging support and gentle guidance, this work, and everything that it represents to me, simply could not exist. More than that, however, my inexpungible debt to Emilio is due to his unique philanthropy, which plucks up individuals like myself (and the incredible people I have called labmates) out of an ocean of heavily credentialed alternatives. It is equally obligatory then, that I thank my first advisor, Zhi-Hong Mao, next. Without Mao, the wild notion of MIT and this life could never have been conceived in the first place, let alone taken root. Mao has also remained equally a mentor and a friend.

My highest honor in this time has been the countenance of my thesis committee members, Dimitris Bertsimas and Patrick Jaillet. They have provided me with all the encouragement (and scrutiny) that a PhD student requires. The title of my thesis is in homage to them, and my imitation is absolutely of the sincerest form of flattery.

Yet my privilege never ends there. I also benefited immeasurably from my triumvirate of mentors: The friendship and guidance of Tom Temple during the early stages of my program were invaluable, in Cambridge, MA, and at Wright Patterson AFB in Dayton, OH. Learning to extract the wisdom from Tom’s peculiar ways was a challenge well worth the effort, for lessons in scientific philosophy and lessons in life. Ketan Savla remains a paragon of intellectual pursuit, whose rare combination of experience, enthusiasm, genius, and patience smoothed the course of all things to be executed for the success of my program; Ketan’s assurance and surgical feedback saw me through my earliest conference proceedings, for which I am especially grateful. His merest utterances are insights worthy to be considered with humility and care. Marco Pavone, my principle co-author, whose unrelenting demand for quality taught me the discipline to write papers correctly; I owe much, also, to the direction and the confidence that Marco showed me.

I would like to thank the people of Singapore (you heard me), for many fascinating experiences. This work was mainly supported, indeed, by the Singapore National Research Foundation, under the Future Urban Mobility SMART IRG program.

My sincerest thanks go to Josh Bialkowski, Kevin Spieser, Pratik Chaudhari, and Sertac Karaman, for countless helpful discussions; to them, and additionally to Michael Otte, Phil Root, and Luis Reyes Castro, for the occasional camaraderie. (To Bethesda Softworks, for my opiates.) Thanks to Mykel Kochenderfer, for an infectious energy, and for showing that there can be lights at the ends of tunnels. A special thanks to Jingjin Yu, whom I have the pleasure recently of calling colleague, for reading through a very long thesis (apparently with a magnifying glass) and providing numerous invaluable improvements.

David and Alison, my parents, have shown tremendous patience for the purgatory of having an only son held away in ivory stasis, seemingly forever. For that they have my love and my everlasting appreciation.

My thesis is dedicated, with all of my devotion, to Darrah.
Contents

1 Introduction ......................................................... 17
  1.1 Motivation ....................................................... 17
  1.2 Context ........................................................... 18
  1.3 The Mathematical Approach in Transport ......................... 19
    1.3.1 Dynamic Vehicle Routing Problems .......................... 19
  1.4 Scope of the Thesis .............................................. 21
  1.5 Contributions and Organization ................................ 21

2 Background Material ............................................... 25
  2.1 Elementary Background ........................................ 25
    2.1.1 Notation ..................................................... 25
    2.1.2 Graph Theory ................................................ 25
    2.1.3 Geometry ..................................................... 26
    2.1.4 Probability Theory: Notation and Basic Results ............. 27
  2.2 Combinatorics ................................................... 28
    2.2.1 Permutations ............................................... 28
    2.2.2 The Traveling Salesman problem (TSP) ...................... 29
    2.2.3 The Stacker Crane problem (SCP) ............................ 30
    2.2.4 The Bipartite Matching problem (BMP) ...................... 31
  2.3 Queueing Models and Control for DVRPs ........................ 32
    2.3.1 Literature Review ........................................... 34
  2.4 Network Optimization (on Graphs) ............................... 36
  2.5 The Earth Mover’s Distance .................................... 36
    2.5.1 Useful Properties .......................................... 37
    2.5.2 Literature Review .......................................... 38

3 Dynamic Taxiing with Generally Distributed Random Demands in 2- and 3- Dimensions ........................................... 41
  3.1 Introduction ....................................................... 41
  3.2 Problem Statement ............................................... 42
  3.3 Previous Work .................................................... 44
  3.4 Preliminaries ..................................................... 46
    3.4.1 Lengths of Large (Random) Stacker Crane tours ............ 46
    3.4.2 Discussion ................................................... 47
  3.5 Routing Policies .................................................. 48
3.5.1 A policy for light load ........................................ 49
3.5.2 A policy for heavy load ........................................ 49
3.6 Performance Analysis .............................................. 49
3.6.1 The SQM Policy .................................................. 49
3.6.2 The Stacker Crane Policy ....................................... 50
3.7 Stability Condition for 1-DPDP .................................. 53
3.8 Performance Lower Bounds ....................................... 55
3.8.1 A light load lower bound ....................................... 55
3.8.2 A heavy load lower bound ...................................... 55
3.8.3 Discussion ...................................................... 59
3.9 Simulation in Heavy-load ......................................... 59
3.9.1 Performance .................................................... 61
3.9.2 Stability Conditions ............................................ 66
3.10 Conclusion .......................................................... 66

4 A Systematic Approach to Fleet-sizing in Practice: A Case Study in Singapore 69
4.1 Introduction .......................................................... 69
4.2 Literature Review: Shared-Mobility Systems .................. 70
4.3 Fleet-Sizing Guidelines for AMoD Systems .................... 71
  4.3.1 Problem Formulation ........................................... 71
  4.3.2 Minimum fleet sizing .......................................... 71
4.4 Data Sources ....................................................... 72
4.5 AMoD Fleet Sizing for Singapore ................................ 73
4.6 Summary of Performance-driven Fleet Sizing .................. 74
4.7 Conclusion and Additional Insights ............................... 75

5 Cost Bounds and Asymptotically Optimal Algorithms for the Euclidean Stacker Crane Problem with Random Demands 77
5.1 Introduction .......................................................... 77
  5.1.1 Contributions ................................................... 77
  5.1.2 Organization .................................................... 78
5.2 Solving the Stacker Crane Problem ............................... 79
5.3 Problem Statement .................................................. 79
5.4 Asymptotically Optimal Polynomial-Time Algorithms for the Stochastic ESCP ........................................ 80
  5.4.1 LARGEARCS and the SPLICE Algorithms ................... 80
  5.4.2 Asymptotic Optimality of LARGEARCS ....................... 82
  5.4.3 Asymptotic Optimality of SPLICE under $Y \perp X$ 84
5.5 Analytical Bounds on the Cost of the ESCP .................... 86
  5.5.1 Lower Bounds on Lengths of Euclidean SCP Tours ......... 86
  5.5.2 Upper Bounds on Lengths of Euclidean SCP Tours ......... 89
5.6 Simulation Results .................................................. 96
  5.6.1 Performance of SPLICE ........................................ 96
  5.6.2 Cost Bounds—First- and Next-Order Asymptotics ........... 97
6 Static and Dynamic Taxiing with Random Demands on Road Networks

6.1 Introduction .............................................. 103
6.2 Road Network Environments .............................. 105
6.2.1 Notation and Representation ......................... 105
6.2.2 The Geometry of Road Networks ..................... 106
6.2.3 A Probability Model for Random Points ............. 108
6.3 Matching Costs on Line Segments ...................... 110
6.4 Matching Costs on Roadmaps ............................ 111
6.5 Cost Bounds for SCP Tours on Roadmaps ............... 113
6.6 Dynamic Taxiing with Generally Distributed Random Demands on Roadmaps ....................................... 114
6.6.1 Analysis of the Stacker Crane Policy in Heavy-load ......................................................... 114
6.6.2 Lower Bounds ........................................... 114
6.6.3 Stability Conditions ................................. 115
6.7 Simulation Study .......................................... 115
6.7.1 Performance of the Stacker Crane policy .......... 115
6.7.2 Stability Conditions ................................. 117
6.8 Conclusion ................................................. 118

7 Fast Bipartite Matching with Roadmap Distances

7.1 Introduction .............................................. 121
7.2 Optimal Matching on Lines and Circles .................. 122
7.3 Problem Statement ....................................... 124
7.4 Optimal Bipartite Matching on a Roadmap ............... 124
7.4.1 Cost Characterization ................................ 124
7.4.2 Cost Bounds on Optimal Matchings ................... 126
7.4.3 Obtaining an Optimal Roadmap Matching ............ 128
7.4.4 Complexity Analysis ................................ 130
7.5 Discussion ................................................. 132

8 An Explicit Formula for the Earth Mover’s Distance with Continuous Road Map Distances

8.1 Introduction .............................................. 135
8.2 Problem Statement ....................................... 137
8.3 The Earth Movers Distance on Road Networks .......... 137
8.3.1 Main Result—One-dimensional Earth mover’s distance .................................................. 137
8.3.2 Alternative (Technical) Formulation ................ 138
8.3.3 Convexity of the EMD Objective ..................... 141
8.3.4 Equivalence of Two Formulations .................... 143
8.4 A Numerical Example ..................................... 144
8.5 Approximating the Earth Movers Distance by Min-Cost Flow .................................................. 145
8.5.1 The General Purpose Scheme ........................ 145
List of Figures

2-1 The two cycles corresponding to the permutation: $\sigma(1) = 3$, $\sigma(2) = 1$, $\sigma(3) = 2$, and $\sigma(4) = 4$. Cycle 1 can equivalently be expressed as $(2, 1, 3)$ or $(3, 2, 1)$. Apart from this cyclic reordering, the decomposition into disjoint cycles is unique. ........................................ 29

2-2 A control system perspective for queueing models. .................. 33

3-1 An illustration of the events which comprise the life-cycle of a single transportation demand. .......................... 43

3-2 Illustration of non-uniform distributions (Cases III and IV), by sampling: $n = 100$ samples for each distribution; origin sites are shown as (red) triangle markers; destination sites are shown as (blue) circles. 60

3-3 Average system time versus workload, in simulation of Case I, under Stacker Crane policy with one unit-speed vehicle. The dashed blue lines has the slope equal to the right-hand side of (3.13) (upper bound). The dashed red line has slope equal to the right-hand side of (3.23) (lower bound). .................................................. 62

3-4 Average system time versus workload, in simulation of Case I, under Stacker Crane policy, with $m = 2$ and $m = 5$ unit-speed vehicles. 63

3-5 Average system time versus fleet size, in simulation of Case I with fixed utilization $\rho = 0.678$, under Stacker Crane policy. .......................... 63

3-6 Average system time versus workload in Case II, under Stacker Crane policy with one unit-speed vehicle. .......................... 64

3-7 Average system time versus workload in Case III, under Stacker Crane policy with one unit-speed vehicle. .......................... 65

3-8 Average system time versus workload in Case IV, under Stacker Crane policy with one unit-speed vehicle. .......................... 65

4-1 Summary of data-derived statistics, used to estimate the minimum fleet size. According to (4.1), the minimum fleet size to serve all of Singapore's mobility demand is 92,693 shared vehicles. At 1,144,400 households in Singapore, that would be roughly one shared car for every 12.3 households. .................. 75

4-2 Average wait times in simulation, as a function of time (periodic), for hypothetical fleet sizes. .................. 76
5-1 Sample execution of a SPLICE algorithm. The solution to the EBMP is \( \sigma(1) = 2, \sigma(2) = 3, \sigma(3) = 1, \sigma(4) = 5, \sigma(5) = 6, \) and \( \sigma(6) = 4. \) Demands are labeled with integers. Origin and destination sites are represented by solid and dashed circles, respectively. Origin-to-destination links are shown as black arrows. Matching links are dark dashed arrows. Subtour connections are shown as lighter, dashed arrows. The resulting tour is \( 1 \to 2 \to 3 \to 6 \to 4 \to 5 \to 1. \)

5-2 Algorithm 2: Demands are labeled with integers. Origin and destination sites are represented by solid and dashed circles, respectively. Origin-to-destination links are shown as black arrows. Shadow origins are shown as dashed squares, with undirected links to their generators (destination sites); also shown are optimal matching links between shadows and origins. Dashed arrows show the resulting induced matching. Note, this solution produces two disconnected subtours \((1, 2, 3)\) and \((4)\).

5-3 Solution (cost) and runtime performance of LARGEARCS and SMALLARCS, versus naive IP solution; twenty-five trials (25) in each of various size categories.

5-4 Illustration of bipartite matching of samples from non-uniform distributions (Cases III and IV): \( n = 100 \) samples for each distribution; origin sites are shown as (red) triangle markers; destination sites are shown as (blue) circles. Plots on the left show samples alone; plots on the right include dashed lines between matched points in the optimal matching.

5-5 Scatter plots of \((n, n^{-1}M^*)\) (top) and \((n, n^{-2/3}(M^* - nM))\) (bottom), with one point for each of twenty-five trials per size category, for Cases III and IV.

6-1 Ratios of shortest path (roadmap) distance versus Euclidean (bird’s eye) distance. Image obtained from [84], produced by randomly sampling a map of Tokyo.

6-2 The simplest geometric network, consisting of a single road.

6-3 A two-road roadmap, with points \( X, Y, \) and \( Z \) demonstrating failure of union metric \( \hat{\mathcal{D}} \) to satisfy triangle inequality.

6-4 A ring network with two roads, \( r_1 \) and \( r_2. \) The shortest path between \( X \) and \( Y \) covers the space to left of \( X \) and \( Y, \) thus using both roads at least partially. The union metric describes the length of the longer path on \( r_2, \) using the space to the right of \( X \) and \( Y. \)

6-5 A square road network with roads: North (N), East (E), South (S), and West (W), all unit length.

6-6 Average system time versus workload under the Stacker Crane policy, with \( m = 1, \) unit-speed vehicle on the square roadmap. Origins and destinations all i.i.d., uniformly distributed.
6-7 Average system time versus fleet size under Stacker Crane policy, with fixed utilization \( p = 0.942 \) under uniformly distributed demand on the square roadmap. ................................................................. 117

6-8 Empirical distribution of average service times (observed vs. predicted) over many random pairs of roadmap and demand distribution. ............. 118

6-9 Trajectories of number of outstanding demands over time \( (T = 10000) \), in stable and unstable regimes (6-9(a) and 6-9(b), respectively). .... 119

7-1 An example road with plot of \( H \); the points in \( S \) denoted by ‘\( x \)’ and the points in \( T \) denoted by ‘o’. ................................................................. 125

8-1 A supply road \( r \in S \). The area \( x \) (under the curve to the left of \( y^* \)) is transported to \( r^- \). The area \( \mu^x(r) - x \) (to the right of \( y^* \)) is transported to \( r^+ \). ................................................................. 139

8-2 A simple road network and the resulting “Wasserstein” flow network. 141

8-3 The roadmap of Figure 8-2(a) labeled with measures \( \mu^x \) and \( \mu^y \). .... 144

8-4 Wasserstein network of the measures in Figure 8-3, labeled with the optimal flow and induced costs. ................................................................. 144

8-5 Bipartite assignment of generated vertices in \( V^x \) and \( V^y \) to the cells in \( C \). ................................................................. 146

8-6 The device \( g_r \) of a supply road \( r \in S \). ................................. 149

8-7 Number of objects instantiated in the Wasserstein network, as a function of the fine-ness parameter \( \epsilon \) of the \( \epsilon \)-tessellation of the roadmap in Figure 8-3. ................................................................. 151

8-8 Quality of the approximation and runtime of the algorithm, versus the fine-ness parameter \( \epsilon \) of the \( \epsilon \)-tessellation of the roadmap in Figure 8-3, to estimate \( W \) between the distributions illustrated in the same figure. The flat line in 8-8(a) indicates performance achieved using \( \mathcal{N}^W \) (the proposed algorithm), which does not depend on \( \epsilon \). ................................. 152
List of Tables

3.1 Compiled statistics for simulated demand models, with estimates of $E_f\|Y - X\|$, $W$, $M$, and $\alpha$. ............................ 61

3.2 Estimated stabilizable rate thresholds $\lambda^*$, for Cases III and IV, with a single unit-speed vehicle. ........................................... 66

5.1 Selected rows from Table 3.1. Compiled statistics for simulated demand models, with estimates of $E_f\|Y - X\|$, $W$, $M$, and $\alpha$. .... 99

6.1 The single-road metric $D_r$ on $r \notin R_d$ (i.e., regular road). Choose $p$ from the first column, and choose $p'$ from the first row. .......... 107

6.2 The single-road metric $D_r$ on $r \in R_d$ (i.e., one-way road). Choose $p$ from the first column, and choose $p'$ from the first row. .......... 107

6.3 Probability mass function (pmf) $\mu(r_1, r_2)$, associated with “toy” demand distribution. ......................................................... 118
Chapter 1

Introduction

This thesis presents new insights and techniques for the design of autonomous or technology-assisted (“intelligent”) transportation systems. The main focus of study is on (i) cooperative control of a fleet of shared transport vehicles, with strict carrying-capacity constraints, and (ii) the on-line setting, where new, random transportation demands enter the system dynamically; that is, demands arrive during the same time in which vehicle assignments and routing decisions are made. The thesis adopts a disciplined and mathematically rigorous approach to identify and seek answers to fundamental questions which remained elusive previously. The thesis also provides a new comprehensive study of the practical considerations of vehicles which must obey “street rules”. Such treatments are surprisingly rare in fundamental research in vehicle routing problems, especially about the on-line (dynamic) and random (stochastic) setting.

1.1 Motivation

Our study is motivated by recent and on-going paradigm shifts in personal and material transportation, enabled by the development and proliferation of networked real-time information systems and autonomous vehicle technology. While such technologies continue to surge forward, less attention has been devoted to the logistics of effectively managing a fleet of potentially thousands of autonomous transport vehicles, or robotaxis. However, as vehicle automation technology matures, planning at all levels will become crucial to its proliferation.

Although a number of existing studies suggest interesting new ideas and operational paradigms for shared vehicle systems, they frequently lack the rigor necessary to justify the feasibility of their claims. Currently, there are very few studies about the potential benefits and limitations of shared vehicle systems with a mathematically rigorous footing. The main goal of this thesis is to borrow insights from a comparatively robust and complementary literature on robotic remote sensing problems, to inform the present study of shared vehicle transportation systems. While the main motivation is provided by automated shared-vehicle systems, the results are applicable to more general cases, including, e.g., a fleet of shared vehicles, each with
a human driver, coordinating with other drivers in order to maximize a quality of service objective.

1.2 Context

*Shared-Vehicle Systems for Efficient Personal Transportation:* Cities face the challenge of maintaining the services and infrastructure necessary to keep pace with the transportation demands of an urban population projected to jump from the current 3.5 billion planet-wide to more than 6 billion in the next 30 years [83]. Private automobiles are an unsustainable solution for the future of personal mobility in such dense urban environments, as the availability of land for road and parking are bound to decrease. Hence, the outdated concept of personal mobility based on private cars is being replaced increasingly by the concept of large-scale vehicle sharing, where people drive (or are driven by) shared vehicles from their points of origin to their destinations:

Traditional vehicle sharing solutions such as taxi services must adapt to meet this challenge, e.g., by mustering increasingly sophisticated systems for information sharing and resource management. At the same time, entirely new paradigms are emerging, such as technology-assisted Mobility-on-Demand (MoD) and Demand-Responsive Transportation (DRT). Under the Mobility-on-Demand paradigm, light and efficient user-driven vehicles are positioned (and periodically repositioned) within an area of dense travel. Under Demand-Responsive Transportation, shuttles follow flexible schedules and routes that may change dynamically in response to user demand.

*Autonomous Vehicles for Personal Mobility:* Research on autonomous vehicles is currently very active [29, 53], and several companies are developing self-driving cars, hoping eventually to replace the human-piloted cars currently on our highways. Proponents of this technology point out a number of benefits, such as (i) increased safety, as the automation reduces the effects of human errors, well known to be the leading cause of traffic accidents, (ii) increased convenience and productivity, as humans are absolved from the more tedious aspects of driving, (iii) increased traffic efficiency and lower congestion, as automated vehicles can precisely monitor one another’s position and coordinate their motion to an extent impossible for human drivers, and (iv) reduced environmental impact, as velocity profiles can be carefully tuned to minimize emissions and noise.

In this thesis, we consider yet another potential benefit, (v) autonomous vehicles as an enabling technology for widespread car sharing. One of the greatest limitations of car-sharing services is the round-trip business model; one-way rental options are uncommon (less viable financially), and often suffer from limited availability. However, if shared cars were able to return to a parking or charging station, or drive to pick up users by themselves, sharing could indeed provide a similar level of convenience as private cars.

*Autonomy for Non-human Transport:* In addition to purely human transportation, the challenge of satisfying the material demands of a growing population is ever increasing, also compounded by the surge of e-commerce and consumer recommenda-
tion systems. However, autonomous vehicles can offer solutions for such challenges as well. In recent years, for example, autonomous vehicles have been incorporated into (i) material handling systems [52], greatly improving supply-chain efficiency, and (ii) hospitals [98], to improve the efficiency and reliability of drug and supply delivery, and to ease the mental burden of logistical concerns on caregivers. Recent and emerging applications for vehicle automation include agriculture [1], and construction.

Vehicle Routing in a Dynamic Setting: A major factor distinguishing the modern transportation settings from past ones is that information is available to decision makers increasingly in real-time, so that such information can be used to improve scheduling and routing efficiency on the fly. Responsible for this increase in the availability of real-time information are the proliferation of wireless communication and networked, automated information systems, especially geographical information systems (GIS). Such technology has tremendous impact whether the vehicles in question are fleet trucks transporting goods from producers to consumers, UAVs surveilling a large area to provide timely updates to information-sensitive decision systems, autonomous cars facilitating the daily commute, or robots constructing human-habitable facilities on other planets (e.g., Mars).

1.3 The Mathematical Approach in Transport

Vehicle routing problems (VRPs) are the mathematical models used to study mobility-based logistics problems. Arguably, the most famous example of a vehicle routing problem is the Traveling Salesman Problem (TSP), where a single vehicle must visit each of a set of target sites, while minimizing the total distance traveled. Like the TSP, some of the earliest and most common VRPs are static combinatorial optimization problems, concerned with large-scale distribution and collection of goods—for a detailed list of common applications, see [122, Chapter 10]. By “static” we mean that all customer demands and their locations—the problem data—are known before vehicle routes are to be determined. In contrast, a dynamic VRP (DVRP) is a vehicle routing problem such that the input data is revealed or updated during the time in which the vehicle executes its routes, allowing for those routes to be updated. As such, DVRPs are useful to study the interaction of transportation systems in real time with information systems. Instead of a static tour or plan, the solution to a DVRP is a policy, encoding the way future information will affect the decision between a number of possible routing contingencies.

1.3.1 Dynamic Vehicle Routing Problems

In the dynamic setting, a routing policy can be thought of mathematically as an online algorithm, that is, an algorithm which processes its input piece-by-piece, in the order that the input is fed to it. In contrast, a static algorithm (or just algorithm) has the entire input available from the start. Online Algorithms (OA) were proposed originally to study combinatorial problems in computer systems design, e.g., memory paging systems [108], multi-processor sharing, etc. Research considering online
algorithms for DVRPs explicitly would begin about a decade later, e.g., with [68].

Even static versions of vehicle routing problems are notoriously challenging and complex combinatorial optimization problems. The dynamic versions, in addition to being inherently more complex, usually need to be solved quickly to produce decisions in real time. A common design philosophy for DVRPs is to solve a related static problem in the initial planning stages, using all information available; then, at future times, e.g., whenever new data is revealed, the solution may be updated using fast heuristics. Examples of such design philosophy can be found in [13, 41, 18].

Evaluation is an important part of policy design: In transportation problems, for example, insight into the influence of system parameters on the performance of systems and routing policies (e.g., the role of fleet size, characteristics of user demand, etc.) can be extremely valuable to the system designer or planner. Simulation study, made feasible by digital computers, has been an effective technique for performance evaluation in many engineering domains. Another approach to evaluation is formal mathematical modeling and analysis. The latter approach has proved to be a crucial aspect of the design of fundamental computer algorithms. However, for formal analysis to be possible, extra discipline is often required in policy design. For example, many of the policies in the cited work above are difficult or impossible to analyze formally.

There are at least two strong arguments for the role of mathematically rigorous performance analysis. First, for safety-critical systems, simulation verification may not be sufficient to ensure that undesired systematic behaviors are eliminated. Second, simulation still leaves us with questions about what are fundamental performance limitations, i.e., what could be achieved by the best policy.

There are two dominant approaches for analyzing DVRPs and the online algorithms (policies) to solve them; i.e., to obtain (i) fundamental performance limitations and (ii) policy-specific performance bounds. The choice of analytical approach indeed can affect the design of policies to obtain strong performance guarantees.

**Competitive Analysis:** Classical treatments of online algorithms generally consider adversarial models of input generation, to obtain best performance in the worst case. The often distinctive feature of this approach, called competitive analysis, was introduced in [69]. The philosophy of competitive analysis is to design and analyze policies for sequential decision making problems which minimize the competitive ratio, or the ratio in the worst case between performance of a causal policy and that of an optimal, offline algorithm, privy to all information a priori. What is remarkable is that even with such a seemingly pessimistic view, for many DVRPs surprisingly good performance can be achieved. In fact, for most DVRP models considered so far, e.g. [68, 66, 85], constant-factor guarantees are common, with factors that are reasonably small. Perhaps the work in this vein most closely related to the thesis is a detailed analysis of the online Dial-a-Ride problem (DARP), appearing in [85].

**Queueing Analysis:** In some cases, though, if there is sufficient historical data to model uncertainty in the demand probabilistically, then the worst-case analysis may be overly pessimistic. For example, in stochastic formulations of many dynamic VRPs, the kinds of instances approaching the competitive ratio can be ruled out with overwhelmingly high probability; that is, those cases tend to be pathological.
This is exactly the perspective motivating a queueing analysis of DVRPs (systems whose inputs include counting processes are often called queueing systems). Given a stochastic process model of demand generation, it is sometimes possible to derive fundamental performance limits for DVRPs using results from queueing theory, or to join results from queueing theory and theory from combinatorial optimization to obtain performance bounds for online algorithms.

1.4 Scope of the Thesis

The focus of the thesis will be on the latter, stochastic and dynamic setting, and on the application and extension of an existing framework (developed originally in [22, 25, 24]) to shared vehicle systems for the transport of physical objects. Such framework has provided numerous insights in the robotics literature on vehicle routing since its inception. We also focus in particular on the so-called One-to-One Pickup and Delivery problems (or, 1-1 PDPs), where individual requests for transportation (demands) entail transport from a specified origin point to a specified destination point. A detailed taxonomy and survey on many other types of PDPs is given in the articles [19] (for static problems) and [20] (for dynamic ones). Since light and inexpensive vehicles are often the favorite, both in the current setting of autonomous vehicular systems and in the evolving transportation setting, we will focus our attention on vehicles with quite low carrying-capacity (even, e.g., for a single parcel or passenger). In contrast, traditional, well-studied large-freight logistics problems have focused mainly on much larger carrying capacities.

1.5 Contributions and Organization

Chapter 2—Background Material: We begin the thesis with a summary of relevant background material: First, we review basic notions from metric geometry and probability theory. Further topics include: combinatorial optimization and an associated probability theory, basic queueing theory and classical spatial extensions, graphical network optimization, and the Earth Mover's distance (EMD).

Part I. The thesis is organized into two parts. The first part of the thesis develops a number of important extensions of the few existing results of queueing analysis for the DVRPs that have three key features: (i) service entailing the transport of physical objects between specific endpoints, (ii) probabilistic uncertainty in the generation of demands, and (iii) vehicles with strict carrying-capacity constraints.

Chapter 3—Dynamic Taxiing with Generally Distributed Random Demands in 2- and 3- Dimensions: In Chapter 3, we introduce several extensions—including, notably, multiple vehicles and general spatial distributions—to a popular idealized model for taxi-like service systems. The main novel contributions of this chapter include: We find a multi-vehicle policy which is provably efficient or nearly-efficient under certain circumstances in heavy demand. (Efficiency is demonstrated by comparing bounds on customer waiting times against policy-independent lower
bounds.) We also give a new checkable condition for stability; that is, to ensure that
the number of unserviced demands can be kept finite for all time by some policy. The
condition is both necessary and sufficient.

The discussion of Chapter 3 actually builds on fairly recent results in a probability
type for pickup and delivery problems; however, we postpone the discussion of those
results until Chapter 5.

Chapter 4—A Systematic Approach to Fleet-sizing in Practice: A Case
Study in Singapore: In Chapter 4 we present part of a recent case study about
personal mobility in the country of Singapore [114]. The study proposes a thought
experiment whereby all modes of personal transport in the city are replaced by a fleet
of shared autonomous vehicles; then, it employs insights from Chapter 3 to consider
the fundamental issue of fleet sizing. Using actual survey data about household
transportation demand, the study suggests that autonomous MoD systems could
meet the personal mobility needs of the entire population of Singapore with a fleet
whose size is approximately a third (1/3) of the total number of passenger vehicles in
operation at the time of this writing. Moreover, combining such fleet-sizing guidelines
with multi-faceted financial analysis, the study makes the case that AMoD systems
are a financially viable alternative to the current (i.e., traditional) means of personal
mobility.

Chapter 5—Cost Bounds and Asymptotically Optimal Algorithms for
the Euclidean Stacker Crane Problem with Random Demands: In Chapter 5,
we identify a class of algorithms for the Euclidean Stacker Crane problem which are
asymptotically optimal, almost surely, under a general probabilistic model. Moreover,
by design, all members of the class run in polynomial time. The algorithms exploit
a key relationship between the Stacker Crane problem and the Bipartite Matching
problem. Second, we provide new asymptotic, probabilistic bounds for the cost of the
optimal bipartite matchings between large random sets of points. The bounds rely
on a connection to a fundamental problem in transportation theory, and its solution,
called the Earth Mover’s distance. We extend the new matching bounds to obtain
bounds on the length of optimal stacker crane tours through large sets of random
demands.

Summary. Ubiquitous among the findings of Part I is that capacity-constrained
vehicles must drive uninhabited (passenger-less) for some non-vanishing fraction of
time. Under probabilistic modelling for the uncertainty of demand, one can predict
the aforementioned fraction precisely, using strong Laws of Large Numbers arguments;
it can be zero only under a very strict condition on the demand distribution, phrased
in terms of the EMD. While related phenomena have been observed under the math-
ematical study of mass transportation theory, to the best of the author’s knowledge,
the application of these concepts to real-time systems is novel. Existing works are
keen to leverage Large Numbers laws to determine inhabited times, but generally fail
to characterize the time spent uninhabited. Since the existence of an unavoidable and
non-negligible extra cost term has significant implications, e.g., for the operational
budget of a transportation service, our results give insight into a phenomenon whose
neglect could prove an unfortunate oversight.

Part II. One of the most significant limitations of analyses in the first part of the
thesis is that they do not consider the “street-rules” that are quite common to ground-based transportation problems; instead, vehicles may travel straight-line distances in the plane. The first part of the thesis ignores such restrictions in order to ease the exposition of the crucial role of the Earth Mover’s distance in analysis. In the second part of the thesis, armed with the resulting framework, we turn our attention to road network-like workspaces.

Chapter 6—Static and Dynamic Taxiing with Random Demands on Road Networks: Many of the results developed in Chapters 3 and 5 have straightforward extensions to road networks. We present these extensions in Chapter 6. A few mathematical preliminaries about road networks and associated probability models are introduced in Section 6.2. Such geometrical and probabilistic machinery is surprisingly difficult to find in standard literature.

Chapter 7—Fast Bipartite Matching with Roadmap Distances: As we will demonstrate in the remaining chapters, several algorithmic improvements are possible on road networks in practice. In Chapter 7, drawing insight from various known results about segments or circles, we present a new algorithm for the Bipartite Matching problem for points on a roadmap, which is efficient in the sense that it achieves the best possible runtime of $O(M \log M)$ to match sets of $M$ points each. With the improved heuristic on roadmaps, one can reduce the runtime of state-of-the-art heuristics for the Stacker Crane problem to match the runtime, e.g., of the fast Christofides’ and minimum-spanning tree heuristics [97], often used to solve TSPs in real or near-real time in the plane.

Chapter 8—An Explicit Formula for the Earth Mover’s Distance with Continuous Road Map Distances: The thesis will have demonstrated a crucial role of the Earth Mover’s distance in analyses of several transportation problems, as one of two terms (a “hidden” term) composing demands’ intrinsic level of “effort”. The EMD also arises in a number of other applications, both within and outside of the transportation context. Few explicit formulas for EMDs have been previously known; the only ones commonly known are for (i) measures over discrete sets, and (ii) measures on a single line segment. In Chapter 8, motivated by a fundamental similarity between the segments of $\mathbb{R}^1$ and the roads of a continuous road network, we derive a new explicit formulation of the Earth Mover’s distance on roadmaps.
Chapter 2

Background Material

In this chapter, we review a number of background notions which are commonly used throughout the thesis. We begin in Section 2.1 with some elementary background such as basic notation, graph theory, as well as fundamentals of geometry, and essential tools of probability theory. Next, in Section 2.2, we introduce several combinatorial problems which are relevant to transportation, including the Traveling Salesman problem (TSP), the Stacker Crane problem (SCP), and the Bipartite Matching problem (BMP). We introduce relevant, preliminary material on the theory and control of queueing systems in Section 2.3, including a brief review of related literature. In Section 2.4, we introduce the Network Optimization problem, which is used extensively throughout the thesis, both for analysis and the design of algorithms. Finally, in Section 2.5, we introduce one of the most central concepts of the entire thesis, a measure of distance between probability distributions commonly known as the Earth Mover’s distance; it turns out that the Earth Mover’s distance plays a crucial role in the laws which govern the throughput of one-way shared vehicle systems.

2.1 Elementary Background

2.1.1 Notation

Let \( \mathbb{Z} \) denote the set of integers, and let \( \mathbb{N}_0 \) and \( \mathbb{N} \) denote the non-negative and positive integers, respectively, i.e., the natural numbers. Let \( \mathbb{R}, \mathbb{R}_{\geq 0}, \) and \( \mathbb{R}_{>0} \) denote the sets of real, non-negative real, and positive real numbers, respectively. Let \( \| \cdot \| \) denote the Euclidean 2-norm.

Asymptotic Notation: For \( f, g : \mathbb{N} \rightarrow \mathbb{R} \), we say that \( f \in O(g) \) (\( f \in \Omega(g) \)) if there exists \( N \in \mathbb{N} \) sufficiently large and \( K \in \mathbb{R}_{>0} \) such that \( |f(n)| \leq K|g(n)| \) (\( |f(n)| \geq K|g(n)| \)) for all \( n \geq N \). If \( f \in O(g) \) and \( f \in \Omega(g) \), then we say \( f \in \Theta(g) \). We say \( f \in o(g) \) if \( \lim_{n \rightarrow +\infty} f(n)/g(n) = 0 \), and \( f \in \omega(g) \) if \( \lim_{n \rightarrow +\infty} |f(n)/g(n)| = +\infty \).

2.1.2 Graph Theory

We assume the reader is familiar with the basic notions in graph theory, e.g., as defined in [45, Ch. 1]. Additionally, we will use the following notation throughout the
thesis: Let \((V, A)\) denote a directed graph, or di-graph, with vertex set \(V\) and a set of directed edges \(A\). In general, \((V, A)\) might be a multi-di-graph, meaning there may be multiple distinct edges between two particular endpoints. For any edge \(a \in A\), we will denote by \(a^-\) the tail of \(a\) (a vertex in \(V\)), and denote by \(a^+\) the head of \(a\). For example, if \(a = (u, v)\), then \(a^- = u\) and \(a^+ = v\).

### 2.1.3 Geometry

**Definition 2.1 (Metric space).** A metric space is the pair of a set \(P\), called a point set, and a distance function \(\mathcal{D} : P \times P \to \mathbb{R}_{\geq 0}\), satisfying for all \(p_1, p_2, p_3 \in P\):

1. \(\mathcal{D}(p_1, p_2) = 0 \implies p_1 = p_2\) (coincidence axiom),
2. \(\mathcal{D}(p_1, p_2) \leq \mathcal{D}(p_1, p_3) + \mathcal{D}(p_3, p_2)\) (triangle inequality), and
3. \(\mathcal{D}(p_1, p_2) = \mathcal{D}(p_2, p_1)\) (symmetry).

(A quasi-metric space is a space that satisfies all of these axioms, except for possibly symmetry. A pseudo-metric space is a space that satisfies all the axioms, except for possibly coincidence.)

Given a sequence of points \(p_1, \ldots, p_n \in P\), let \(\mathcal{D}(p_1, \ldots, p_n) = \sum_{k=1}^{n-1} \mathcal{D}(p_k, p_{k+1})\), i.e., it is the length of the “walk” through each point in the order of the arguments.

**Definition 2.2 (Curves).** A curve, or path, is a continuous mapping \(P : I \to P\), from a non-empty interval \(I\) of the real line to \(P\). (Continuity here is in the \(\epsilon/\delta\) sense, i.e., for every \(s \in I\) and \(\epsilon > 0\), there exists \(\delta > 0\) such that \(\mathcal{D}(P(s), P(t)) \leq \epsilon\) for every \(t \in I\) such that \(|t - s| < \delta\).)

We will call the restriction of a curve or path \(P : I \to P\) to some sub-interval of \(I\) a fragment of \(P\).

**Definition 2.3 (Connectedness).** In this thesis, we will say a metric space \((P, \mathcal{D})\) is connected if \(\mathcal{D}(p_1, p_2) < +\infty\) for all \(p_1, p_2 \in P\).

**Definition 2.4 (Path connectedness).** A metric space \((P, \mathcal{D})\) is said to be path connected if for any two points \(p_1, p_2 \in P\) there exists a curve \(P\) such that \(P(0) = p_1\) and \(P(1) = p_2\).

Note that connectedness and path connectedness are not necessarily equivalent.

**Definition 2.5 (Length).** We will define the length of a curve in the sense of rectifiable curves. Namely, given a curve \(P : I \to P\) and a finite sequence of coordinates \(s_1 \leq s_2 \leq \ldots \leq s_n\), with all \(s_k \in I\), we can define the sum \(\sum_{k=1}^{n-1} \mathcal{D}(P(s_k), P(s_{k+1}))\). The arc length of \(P\) is defined as the supremum of this sum among possible sequences, with \(n\) unbounded.
Definition 2.6 (Intrinsic metric). Given a path connected metric space $(P, D)$, we can define a distance function $D_1$, where $D_1(p_1, p_2)$ is the infimum of lengths among all curves from $p_1$ to $p_2$. The resulting metric space $(P, D_1)$ is called a length space. (If $D$ and $D_1$ coincide everywhere, then $(P, D)$ is equally a length space.)

Definition 2.7 (Distance graph). Given a metric space $(P, D)$, the distance graph $(Q, A)$ on a finite subset of points $Q \subset P$ is the pairing of $Q$ with a fully connecting, weighted edge set $A = Q^2$, where every edge $a \in A$ has weight $D(a^-, a^+)$. 

2.1.4 Probability Theory: Notation and Basic Results

We assume the reader is familiar with the basic notions of probability theory as covered in an introductory graduate course on the subject. For two events $E_1$ and $E_2$, we will denote by $P[E_1]$ the probability of the event $E_1$, and by $P[E_2 | E_1]$ the probability of $E_2$ conditioned on $E_1$. We will use capital letters to denote random variables wherever possible. For a random variable $X : \Omega \to \mathbb{R}$ (where $\Omega$ denotes the set of possible outcomes), we will denote by $E[X]$ the expectation of $X$ and by $E[X | E]$ the expectation of $X$ conditioned on event $E$. We will denote by $\text{Var}[X] = E[(X - E[X])^2]$ the variance of $X$.

Common inequalities: The Markov inequality states that for any non-negative random variable $X$ and $a > 0$, $P[X \geq a] \leq E[X]/a$. The Chebyshev inequality states that $P[|X - E[X]| \geq a] \leq \text{Var}[X]/a^2$. Jensen’s inequality states that, given a real-valued random variable $X$ and a convex, real-valued function $\varphi$, that $\varphi(E[X]) \leq E[\varphi(X)]$.

Convergence of random variables: A sequence of random variables $\{X_i : i \in \mathbb{N}_0\}$ converges “almost surely” (a.s.) to a random variable $X$ if $P\{\omega : \lim_{i \to +\infty} X_i(\omega) = X(\omega)\} = 1$. The convergence is “sure” if $\lim_{i \to +\infty} X_i(\omega) = X(\omega)$ for every $\omega \in \Omega$. The sequence converges “in probability” or “with high probability” if $\lim_{i \to +\infty} P[|X_i(\omega) - X(\omega)| > \epsilon] = 0$ for every $\epsilon > 0$.

Theorem 2.8 (The Strong Law of Absolute Differences). Let $X_1, \ldots, X_n$ be a sequence of scalar random variables that are i.i.d. with mean $E[X]$ and finite variance. Then the sequence of cumulative sums $S_n = \sum_{i=1}^n X_i$ has the property (discussed, e.g., in [16]) that

$$\lim_{n \to \infty} \frac{S_n - E S_n}{n^\alpha} = 0, \quad \text{almost surely,}$$

for any $\alpha > 1/2$. The famous Strong Law of Large Numbers (SLLN) is the special case where $\alpha = 1$.

Measure-theoretic treatment: The probability models defined for road networks starting in the second part of the thesis in Chapter 6 will require additional rigor from the measure-theoretic framework of probability spaces, presented, e.g., in [12, Ch. 6]. A probability distribution $\mu$, e.g., on the Borel sets of $\Omega = \mathbb{R}^d$ for some $d \geq 1$, is absolutely continuous with respect to the Lebesgue measure $\lambda$ if there exists a function $\varphi : \Omega \to \mathbb{R}$ such that $\mu(\mathcal{E}) = \int_{\mathcal{E}} \varphi \, d\lambda$ for any event $\mathcal{E}$. 

27
2.2 Combinatorics

2.2.1 Permutations

Many combinatorial problems are concerned with optimal orderings of sets (e.g., of tasks), and permutations provide the mathematical formalism of orderings. A permutation is a rearrangement of the elements of an ordered set \( S \) according to a bijective correspondence \( \sigma : S \rightarrow S \). As an example, a particular permutation of the ordered set \( (1, 2, 3, 4) \) is \( \sigma(1) = 3, \sigma(2) = 1, \sigma(3) = 2, \) and \( \sigma(4) = 4 \), which leads to the reordered set \( (3, 1, 2, 4) \). The number of distinct permutations on a set of \( n \) elements is given by \( n! \) (factorial). We denote the set of permutations over the \( n \)-element ordered set \( (1, \ldots, n) \) by \( \Pi_n \). We will use the following elementary properties of permutations, which follow from the fact that permutations are the bijective correspondences:

**Property 1** There is an *identity permutation* \( \sigma_1 \) which maps every element of a set \( S \) to itself.

**Property 2** Given two permutations \( \sigma, \sigma' \in \Pi_n \), the composition \( \sigma \sigma' \) is again a permutation.

**Property 3** Each permutation \( \sigma \in \Pi_n \) has an inverse permutation \( \sigma^{-1} \), with the property that \( \sigma(x) = y \) if and only if \( \sigma^{-1}(y) = x \). (Thus, note that \( \sigma^{-1} \sigma = \sigma_1 \).)

**Property 4** For any \( \bar{\sigma} \in \Pi_n \), it holds \( \Pi_n = \{ \sigma \bar{\sigma}, \sigma \in \Pi_n \} \); in other words, for a given permutation \( \bar{\sigma} \) playing the role of basis, \( \Pi_n \) can be expressed in terms of composed permutations.

A permutation can be conveniently represented in a two-line notation, where one lists the elements of \( S \) in the first row and their images in the second row, with the property that a first-row element and its image are in the same column. For the previous example, one would write:

\[
\begin{bmatrix}
1 & 2 & 3 & 4 \\
3 & 1 & 2 & 4
\end{bmatrix}
\]  \hspace{1cm} (2.1)

The inverse permutation can be obtained by switching the rows of this notation (and re-ordering the columns).

A permutation \( \sigma \in \Pi_n \) is said to have a *cycle* \( \mathcal{W} \subseteq S \) if the objects in \( \mathcal{W} \) form an orbit under the system \( \ell_{i+1} = \sigma(\ell_i) \), i.e., starting the system from anywhere in \( \mathcal{W} \), the set of all elements reached is also \( \mathcal{W} \). Each permutation represents a *unique* partition of \( S \) into disjoint cycles, apart from cyclic reordering of the elements within each cycle (see Figure 2-1). Henceforth, we denote by \( N(\sigma) \) the number of distinct cycles of \( \sigma \). In the example in equation (2.1), there are two cycles, namely \( (1, 3, 2) \), which corresponds to \( \sigma(1) = 3, \sigma(3) = 2, \sigma(2) = 1 \), and \( (4) \), which corresponds to \( \sigma(4) = 4 \) (see Figure 2-1).

A useful probabilistic result is given in [105], about the number of cycles of a random permutation:
Figure 2-1: The two cycles corresponding to the permutation: \( \sigma(1) = 3, \sigma(2) = 1, \sigma(3) = 2, \) and \( \sigma(4) = 4 \). Cycle 1 can equivalently be expressed as \((2, 1, 3)\) or \((3, 2, 1)\). Apart from this cyclic reordering, the decomposition into disjoint cycles is unique.

**Proposition 2.9.** Suppose that all elements of \( \Pi_n \) are assigned probability \( \frac{1}{n!} \), i.e.,

\[
\mathbb{P}[\sigma] := \mathbb{P}[\text{One selects } \sigma] = \frac{1}{n!}, \quad \text{for all } \sigma \in \Pi_n.
\]

The number of cycles \( N(\sigma) \) of \( \sigma \) has expectation and variance both \( \log(n) + O(1) \); here \( \log \) denotes the natural logarithm.

### 2.2.2 The Traveling Salesman problem (TSP)

**Definition 2.10 (Traveling Salesman problem (TSP)).** Given a finite set \( X \) of \(|X|\) points from a metric space \((P, D)\), the Traveling Salesman problem is to find the minimum-length tour (usually, cyclic) which visits each point in \( X \) [at least] once.

We will denote by \( \text{TSP}(X) \) the set of tours which satisfy the so-called TSP constraints, i.e., they visit every point in \( X \). We will denote by \( \text{TSP}^*(X) \) the optimal or minimum length such tour. Depending on context, the above notation may refer instead to the lengths of such tours, or to the order in which they visit the points in \( X \). Since these three meanings lie in distinct mathematical categories (i.e., paths versus lengths versus permutations), we can always rely on context to resolve our abuse of notation. For example, by convention, \( \text{TSP}^*(\emptyset) = 0 \); according to context, this usage of \( \text{TSP}^* \) concerns the length of the optimal tour through an empty set.

If the points are in \( \mathbb{R}^d \) with the 2-norm distance \( \| \cdot \| \), for some \( d \geq 1 \), then we call such problem the *Euclidean* TSP, or ETSP. The TSP is an NP-Hard problem, both in general and in the Euclidean case.

**Probability theory for the ETSP**

The TSP has some interesting properties on sets of random points in Euclidean space.

**Theorem 2.11 (Beardwood, Halton, Hammersley [17]).** There is a constant \( \beta_{\text{TSP},d} \) [for each \( d \geq 1 \)] such that for any sequence of independent random variables \( \{X_i\} \) with distribution \( \mu \) of compact support [on \( d \)-dimensional Euclidean space], we have

\[
\lim_{n \to +\infty} \frac{\text{ETSP}^*(X_1, X_2, \ldots, X_n)}{n^{(d-1)/d}} = \beta_{\text{TSP},d} \int_{\mathbb{R}^d} \varphi(x)^{(d-1)/d} \, dx, \quad \text{a.s.,} \quad (2.2)
\]
where $\varphi$ is the density of the absolutely continuous part of $\mu$.

It has been shown [94] that $\beta_{\text{TSP}, 2} = 0.7120 \pm 0.0002$ for $\mathbb{R}^2$ and $\beta_{\text{TSP}, 3} = 0.6972 \pm 0.0002$ for $\mathbb{R}^3$.

In the future, we will use a convention for shorthand notation where $\text{TSP}_\mu(n)$ denotes the TSP through $n$ points independently, identically distributed (i.i.d.) with distribution $\mu$. It is known (e.g., from [116, Sec. 2.3]) that

$$\lim_{n \to +\infty} \mathbb{E} \left[ \frac{\text{ETSP}_\mu(n)}{n^{(d-1)/d}} \right] = \beta_{\text{TSP}, d} \int \varphi(x)^{(d-1)/d} dx.$$ 

While not unexpected given the former result, the expectation limit is not strictly implied by the theorem. The ETSP is known to have a worst-case (non-probabilistic) upper bound of the same order as (2.2) (see, e.g., [115]).

The so-called BHH theorem has been crucial in previous studies of vehicle routing problems under probabilistic uncertainty. It highlights an important property of TSP tours: that the average tour length per point on the tour $(n^{-1} \text{ETSP}_\mu(n))$ vanishes as $n$ grows large. For vehicle routing problems, this property allows that the costs due to travel between locations of interest become negligible as the number of locations grows large.

### 2.2.3 The Stacker Crane problem (SCP)

The Stacker Crane problem (SCP) is a less famous cousin of the TSP. It is discussed in [19] and [57].

**Definition 2.12 (Stacker Crane Problem).** Given ordered sets $\mathcal{X} = \{X_i\}_{i=1}^n$ and $\mathcal{Y} = \{Y_i\}_{i=1}^n$, of $|\mathcal{X}| = |\mathcal{Y}| = n$ points each from a metric space $(P, \mathcal{D})$, the Stacker Crane problem is to find the minimum length tour which, for some permutation $\sigma$ of the natural numbers $\{1, 2, \ldots, n\}$, visits all points at least once, in the order $X_{\sigma(1)}, Y_{\sigma(1)}, X_{\sigma(2)}, Y_{\sigma(2)}, \ldots, X_{\sigma(n)}, Y_{\sigma(n)}$.

The points $\{X_i\}$ are called *origins* and the points $\{Y_i\}$ are called *destinations*. The SCP models the problem of transporting $n$ demands or distinct objects in space, one at a time, where the $i$th demand is collected at the point $X_i$ and must be delivered to the point $Y_i$. Thus, the points $X_i$ and $Y_i$ are often grouped together: the pairs $\{(X_i, Y_i)\} =: \mathcal{D}$ are called the origin-destination pairs, or O/D pairs. (The SCP is sometimes known as the “Unit-capacity, One-to-one Pickup and Delivery Problem”.)

We will denote by $\text{SCP}(\mathcal{D})$ the set of tours which satisfy the SCP constraints with respect to the set of origin-destination pairs $\mathcal{D}$; given the ordered sets $\mathcal{X}$ and $\mathcal{Y}$ instead, we may write $\text{SCP}(\mathcal{X}, \mathcal{Y})$ instead. We will denote by $\text{SCP}^*(\mathcal{D})$ the optimal or minimum length stacker crane tour. Again, depending on context, the notation may refer instead to the lengths of stacker crane tours or to the order in which they engage the pairs.

We will use a convention of notation where $\text{SCP}_f(n)$ denotes the SCP through the first $n$ pairs of an i.i.d. random sequence $\{(X_i, Y_i)\}$ with distribution $f$ over $\Omega \times \Omega$, i.e., $X_i$ and $Y_i$ may be jointly distributed.
In the case that points are in $\mathbb{R}^d$ with the 2-norm distance, for some $d \geq 1$, then we call such problem the **Euclidean** SCP, or ESCP. Note that if we co-locate the origins and destinations, i.e. $Y_i = X_i$ for all $i$, then we recover the TSP. Thus, the SCP is also NP-Hard in the both the general and Euclidean cases.

### 2.2.4 The Bipartite Matching problem (BMP)

**Definition 2.13 (Bipartite Matching problem (BMP)).** Given sets $\mathcal{X}$ and $\mathcal{Y}$, of $|\mathcal{X}| = |\mathcal{Y}| = n$ objects each, and a cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the Bipartite Matching problem (BMP), or Assignment problem (AP), is to find a bijective mapping $\sigma : \mathcal{X} \to \mathcal{Y}$ such that the total cost $\sum_{X \in \mathcal{X}} c(X, \sigma(X))$ is minimized.

We call a feasible mapping a *matching* or *assignment*, and we refer to the tuples $(X, \sigma(X)) : X \in \mathcal{X}$ as *matches*. The metric BMP results when $\mathcal{X}$ and $\mathcal{Y}$ are points chosen from a metric space $(\mathcal{P}, \mathcal{Q})$ and $c(X, Y) = \mathcal{Q}(X, Y)$ everywhere on $\mathcal{X} \times \mathcal{Y}$. The thesis focuses almost entirely on the metric case. When $\mathcal{P} = \mathbb{R}^d$ $(d \geq 1)$ and $\mathcal{Q}$ is the 2-norm $\| \cdot \|$, then we call it the **Euclidean** Bipartite Matching problem (EBMP).

In the metric case, we denote by $M(\mathcal{X}, \mathcal{Y})$ the set of assignments whose total cost is finite; we call these *feasible* assignments. We denote by $M^*(\mathcal{X}, \mathcal{Y})$ the optimal or minimum cost assignment. Depending on context, the notation may also refer instead to the cost itself.

#### Probability Theory for the EBMP

The EBMP over random sets of points enjoys some remarkable properties:

**Theorem 2.14 (Dobric-Yukich High-dimensional Matching [46]).** There is a constant $\beta_{M,d}$ [for each $d \geq 3$], such that for any sequence of independent random variables $\{Z_i\}$, with distribution $\mu$ of compact support [over $\mathbb{R}^d$], and odd and even subsequences $\{X_i\}$ and $\{Y_i\}$ respectively, the optimal bipartite matching cost $M^*(X_1, \ldots, X_n, Y_1, \ldots, Y_n) = \min_{\sigma \in \Omega_n} \sum_{i=1}^n \| X_{\sigma(i)} - Y_i \|$ has limit behavior

$$
\lim_{n \to +\infty} \frac{M^*(X_1, \ldots, X_n, Y_1, \ldots, Y_n)}{n^{(d-1)/d}} = \beta_{M,d} \int_{\Omega} \varphi(x)^{(d-1)/d} \, dx, \tag{2.3}
$$

almost surely, where $\varphi$ is the density of the absolutely continuous part of the point distribution $\mu$.

It has been shown [62] that $\beta_{M,3} = 0.7080 \pm 0.0002$. Unfortunately, in $\mathbb{R}^2$ (the planar case), only the following weaker result holds:

**Theorem 2.15 ([120]).** With high probability as $n \to +\infty$ (i.e. with probability $1-o(1)$):

$$
\frac{M^*(X_1, \ldots, X_n, Y_1, \ldots, Y_n)}{(n \log n)^{1/2}} \leq \gamma \tag{2.4}
$$

for some positive constant $\gamma$. 

31
If the probability distribution is uniform, it also holds with high probability that 
\[ M^*(X_1, \ldots, X_n, Y_1, \ldots, Y_n)/(n \log n)^{1/2} \] is bounded below by a positive constant [5].

In this thesis, we will turn our attention to a case where the sequences \( \{X_i\} \) and 
\( \{Y_i\} \) are i.i.d. and distributed according to distributions \( \mu_X \) and \( \mu_Y \), respectively, 
which may be distinct. Curiously, to the best of the author’s knowledge, there have 
been no previous analytical studies for this more general case. In the future, we will 
use a convention for shorthand notation where \( M(n; \mu_X, \mu_Y) \) denotes the BMP on the 
first \( n \) points in each set; we will write simply \( M(n; \mu) \) if \( \mu = \mu_X = \mu_Y \).

There are also non-asymptotic bounds for the expectations of matchings (see [46]): 
For example, the derivation of Ajtai et. al. [5], in the setting \( d = 2 \), essentially shows 
that there is a constant \( C \) such that 
\[ E[M^*(n; f)]/(n \log n)^{1/2} \leq C. \] (2.5)

Dobric and Yukich [46] showed there is a constant \( C \) such that for \( d \geq 3 \),
\[ \frac{E[M^*(n, f)]}{n^{(d-1)/d}} - \alpha \leq \frac{C}{n^{(d-2)/(d(d+2))}}, \] (2.6)
where \( \alpha \) is the right-hand side of (2.3). The size of the constant \( C \) may depend on 
the distribution of points, however, a uniform bound can be obtained which holds for 
all distributions, because the uniform distribution of points obtains the worst case in 
general.

\section*{Solving the Bipartite Matching problem}

The general version of the Bipartite Matching problem is solved by the “Hungarian” 
method [73], which runs in \( O(n^3) \) time. Alternatively, the assignment problem can be 
cast as a linear program (LP), which can be solved generically. Most algorithms for 
the Euclidean case rely on the Hungarian method at some point. However, the \( O(n^3) \) 
barrier was indeed broken by Agarwal et. al [3], who presented a class of algorithms 
running in \( O(n^{2+\epsilon}) \), where \( \epsilon \) is an arbitrarily small positive constant. Additionally, 
there are also several approximation algorithms: among others, the algorithm pre-
seated in [2] produces a \( O(\log(1/\epsilon)) \) optimal solution in expected runtime \( O(n^{1+\epsilon}) \), 
where, again, \( \epsilon \) is an arbitrarily small positive constant.

\section*{2.3 Queueing Models and Control for DVRPs}

We assume that the reader is familiar with the basic notions of Queueing Theory, 
e.g., as discussed in [72]. The early pioneering work applying queueing models to 
DVRPs was a sequence of papers on the Dynamic Traveling Repairman Problem 
(DTRP) [22, 25, 24]. The distinguishing feature of the DTRP is that the servers 
must travel to the sites of demands to render service; the demands are randomly 
spatially distributed within a bounded Euclidean workspace.
Out of this initial study, a diverse study of DVRPs incorporating queueing elements was born; for a brief literature review, see Section 2.3.1. As the kinds of queueing scenarios under consideration become more various and complex, it is perhaps useful to maintain a taxonomy of these problems. To facilitate such classification, it will be useful to identify significant model components. In this thesis, we will approach the compartmentalization of queueing models using a perspective common to control theory:

![Diagram of a control system perspective for queueing models.](image)

A complete (autonomous) queueing system generally comprises (i) a stochastic process driving the generation of demands, (ii) a system of rules governing the way in which one may interact with demands using a set of resources (e.g., vehicles) in a specified environment, and (iii) a policy or service discipline for fulfilling demands according to such rules. These components are mutually related in a way quite similar to the way in which the exogenous input, plant, and controller, respectively, are related in a traditional control system model (see Figure 2-2).

In the thesis we will refer to the combination of all elements except the policy as a theater. As an example, we will now introduce a slightly generalized model of the theater of study under the DTRP; we will call it the **homogeneous repairman theater**:

**Homogeneous repairman theater**: Demands arrive into the system according to a homogeneous (time-invariant) Poisson process with time intensity $\lambda$. The $i$th demand requires first that a point $X_i$ be visited by some vehicle. The points $\{X_i\}$ are i.i.d., with $X_i \sim f$. All arrived demands are known immediately, including their
locations. Once a vehicle reaches $X_i$, it remains on site to provide some service for time $s_i \geq 0$. The service times $s_i$ are i.i.d., with $s_i \sim \mu$, but are not known a priori. Once the service time has elapsed, the demand is removed from the system and the vehicle is freed. A homogenous fleet of $m$ vehicles move at maximum speed $v$ in an environment $(\mathcal{P}, \mathcal{D})$. The vehicles have infinite range and servicing capacity, but they must remain within a compact workspace $\mathcal{W} \subset \mathcal{P}$. (We assume $f$ is supported entirely on $\mathcal{W}$.) The homogenous repairman theater is specified by the tuple $(\lambda, (\mathcal{P}, \mathcal{D}), f, \mu, W, m, v)$.

Queueing Statistics: The performance of policies can be related to some common metrics: The system time $T_i$ of demand $i$ is defined as the time elapsed from the arrival of demand $i$ to the time it is removed from the system. The waiting time $W_i$ is defined as the time elapsed from arrival of demand $i$ to the time it became engaged in service. In the repairman theater, the waiting period of demand $i$ ends once the vehicle reaches $X_i$, i.e., $T_i = W_i + s_i$. The steady-state system time is defined, given a stationary, task allocation and vehicle routing policy $\pi$, by $\bar{T}_\pi \doteq \limsup_{i \to +\infty} E[T_i | \pi]$.

**Definition 2.16 (Stability).** Given a theater $\mathcal{M}$, a routing policy $\pi$ is said to be stabilizing if $\bar{T}_\pi < +\infty$, i.e., the expected system time remains uniformly bounded for all times. Then the system $(\mathcal{M}, \pi)$ is said to be stable. If the set of policies $\{\pi \in \Pi | \bar{T}_\pi < +\infty\}$ is not empty, i.e., $\bar{T}^* < +\infty$, then $\mathcal{M}$ is stabilizable.

**Problem 2.1 (Dynamic [and Stochastic] Traveling Repairman problem).** Consider the homogeneous repairman theater with $m$ unit speed service vehicles, and let $\Pi$ denote the set of all causal, stationary, stabilizing routing policies. The Dynamic Traveling Repairman problem (m-DTRP) is to find a policy $\pi^* \in \Pi$ (if one exists) such that $\bar{T}_{\pi^*} = \inf_{\pi \in \Pi} \bar{T}_\pi \doteq \bar{T}^*$.

Among the results of [24] were (i) conditions, both necessary and sufficient, to ensure the existence of routing policies to keep the number of unserviced demands in the system bounded; (ii) policy-independent lower bounds on the time spent in the system by an average customer in steady state; (iii) an optimal policy in light-load, i.e., for near-zero rate of arrivals; and, (iv) routing policies for heavy-load, i.e., for near-critical rate of arrivals, which have average system time within a constant factor of the lower bound.

### 2.3.1 Literature Review

**Literature Review—The Dynamic Traveling Repairman Problem**

Although initially a study within the Operations Research community, the study of the stochastic DTRP paved the way for an expansive study of DVRPs with a strong focus in robotic planning, about a decade later. To give an idea of how diverse and robust are the results in this study, we provide an annotated list of works below:

- **Scalable, adaptive, distributed control:** An early extension [55] focused on achieving the stability and performance guarantees of the pioneering work, but with scalable policies that are spatially distributed, and adaptive to changes in the environment (making such policies more attractive to large-scale system designers). In [10] it is
shown that efficient multi-vehicle policies exist which are adaptive and make little or no explicit communication (either with a centralized authority, or other vehicles). In [88], an adaptive, distributed algorithm for geometric partitioning is presented which allows vehicles to converge to an optimal distribution of “workload”; that algorithm forms the basis of an efficient, adaptive, and distributed algorithm for multi-vehicle DTRP in [90].

**Dynamical vehicles:** A highly non-trivial extension of the theory was developed in [49, 51], studying the effect of more realistic models of vehicle dynamics; in particular, Dubins’s bounded-curvature vehicle, as a more accurate model of fixed-wing UAVs and/or steered ground vehicles that are ubiquitous in commercial and military mobility settings. The work relied heavily on new algorithms for the TSP with Dubins’s vehicles [104]; interestingly, performance limits and algorithms for the Dubins’s DTRP remain open questions if targets are not uniformly distributed in the workspace. (As in previous work, light- and heavy-load performance bounds were given, as well as efficient policies, first for the single-vehicle [49], then the multi-vehicle case [51].)

**Persistent Patrol, surveillance:** The DTRP was augmented with a search component in [50], i.e. target arrivals and locations are not revealed, but must be discovered by agents in order to be scheduled for “service”; this produces a more appealing model for a dynamic surveillance scenario; [63] revisits the problem (now refered to as the *persistent patrol problem* (PPP)) in the case of general (non-uniform) distribution of targets. A “persistent monitoring” problem is presented and analyzed in [113], where a large number of spatially distributed features require service repeated ad infinitum (e.g. monitoring); each feature is endowed with an *urgency* that grows as the time elapsed since last service, and vehicles move to keep all urgencies bounded.

**Time windows:** A DTRP with customer impatience (time-windows) was considered in [89], where it was determined the number of vehicles needed to ensure a high probability of punctual customer service, as well as a spatially distributed policy with strong performance guarantees.

**Heterogeneity:** A flavor of the DTRP with priority classes for customers was developed in [111]; the authors compare the development to that of the classical theory of priority queues [61, Chapter 16]. The *dynamic team forming problem* (DTFP) is developed in [109], an extension of the DTRP in which demands are *tasks* that require multiple services for completion, which are provided by a team of heterogenous vehicles, each providing a subset of the total services available.

**Game-theoretic models:** In [103], the authors present a game-theoretic mechanism whose Nash equilibria correspond to desirable agent configurations in the light-load case. The papers [110, 26] treat a *pursuit-evasion* twist on DTRP where demands appear randomly on a segment and then travel at some speed in a given perpendicular direction: [110] models a boundary defense scenario (the agent’s objective is to service targets before they reach a given boundary); [26] is an “escape-and-capture” version of the problem; targets appear and move at unit speed in a given direction; the agent moves to service targets in order to keep the total number of targets small (e.g. uniformly bounded in expectation). A DTRP-based zero-sum game is presented in [54], which resembles a counter-terrorism scenario; a motion-constrained adversary
places targets (e.g., improvised explosive devices) into the environment from a depot “stash”, and the service vehicle attempts to minimize the average target waiting time.

2.4 Network Optimization (on Graphs)

Network Optimization is part of a classical theory of network flows with applications in most branches of engineering and mathematics. Although we give here only essential definitions of the Network Optimization problem, a very thorough treatment of theory, applications, and algorithms can be found in [4].

An input to the Network Optimization problem consists of a flow network $N$ and edge costs $c$. Here, we will develop the formulation where a flow network is the combination of a di-graph $(V, A)$ and a supply mapping $b : V \rightarrow \mathbb{R}$. The mapping $b$ associates with every vertex $u \in V$ a supply $b(u)$; if $b(u) > 0$, then $u$ is called a supply node; if $b(u) < 0$, then $u$ is called a demand node, with “demand” $-b(u) > 0$; if $b(u) = 0$, then $u$ is called a transshipment node. (We assume that $\sum_{u\in V} b(u) = 0$.)

A flow on $A$ is any non-negative mapping $f : A \rightarrow \mathbb{R}_{\geq 0}$. An admissible flow on $N$ is a flow on $A$ satisfying

$$b(u) + \sum_{a \in A : a^+=u} f(a) = \sum_{a \in A : a^-u} f(a) \quad (u \in V). \quad (2.7)$$

We call (2.7) the flow conservation constraints. We use standard shorthand notation $f \in N$ (e.g., see [4]) to say $f$ is admissible by flow network $N$.

The edge costs $c$ provide flows on $N$ with a notion of cost. We will interpret $c$ as a collection of mappings $\{c(\cdot ; a) : a \in A\}$, and we define the total cost of a flow $f \in N$ [under edge costs $c$] as

$$J(f; c) \doteq \sum_{a \in A} c(f(a); a). \quad (2.8)$$

If edge costs $c$ have the property that $J(f; c)$ is linear in $f$, i.e., for some edge weights $w : A \rightarrow \mathbb{R}_{\geq 0}$, $J(f; c) = \sum_{a \in A} w(a)f(a)$, then we write $J(\cdot; w) \equiv J(\cdot; c)$.

The network optimization problem is to find a minimum-cost admissible flow, i.e., a flow $f \in N$ such that $J(f; c) \leq J(f'; c)$ for all $f' \in N$; note that this is a finite-dimensional optimization problem with only linear equality and inequality constraints.

2.5 The Earth Mover’s Distance

The Earth mover’s distance (EMD) is a measure of distance between probability distributions—or measures, more generally—which is commonly encountered in mathematics and computer science. In mathematics, it is generally referred to as the Rubenstein/Kantorovich/Wasserstein distance, or simply Wasserstein distance [102]. The metric is also the solution to the Monge-Kantorovich problem, which is at the
heart of mass transportation theory \cite{95, 96}. A common informal interpretation of the EMD is that if one treats two measures (say, $\mu^a$ and $\mu^b$) as two distinct ways of arranging some fluid/continuous commodity (e.g., “a pile of dirt”) in a spatial domain $\Omega$, then the EMD is the minimum cost of transforming the arrangement described by $\mu^a$ into the arrangement described by $\mu^b$. Such interpretation requires that the underlying domain be equipped with a “ground metric” $\mathcal{D}: \Omega \times \Omega \to \mathbb{R}_{\geq 0}$ by which the cost of transformations can be measured; the notion is that relocating a unit of commodity from a point $p \in \Omega$ to point a $p' \in \Omega$ incurs cost $\mathcal{D}(p, p')$. Formally, the EMD is defined, given a complete and separable metric space $(\Omega, \mathcal{D})$ as

$$\mathcal{W}(\mu^a, \mu^b) = \inf_{\gamma \in \Gamma(\mu^a, \mu^b)} \int_{\Omega} \mathcal{D}(p, p') \, d\gamma(p, p').$$

The search space $\Gamma$ is the set of couplings of $\mu^a$ and $\mu^b$, i.e., the collection of all joint measures over $\Omega^2$ having marginals $\mu^a$ and $\mu^b$ on the first and second factors, respectively. Generally speaking, $\Gamma$ is infinite-dimensional. If the domain is $\mathbb{R}^d$, $d \geq 1$, and the ground metric is the $2$-norm $\| \cdot \|$, then we will call it the Euclidean EMD.

There are some known connections between the Earth Mover’s distance and the metric Bipartite Matching problem. For example, it is known that the cost of the optimal bipartite matching between two point sets is equal to the Earth Mover’s distance between their empirical distributions. Thus, it is fairly intuitive that the EMD can predict the costs of metric bipartite matchings between large random sets through Laws of Large Numbers arguments. We will formalize such notions in Chapter 5. Such connections are exploited throughout the thesis, both for the analysis and the design of algorithms.

### 2.5.1 Useful Properties

When the domain $\Omega$ is a finite set, then the EMD is given by the cost of the optimal solution to:

**Problem 2.2 (EMD, discrete).** Minimize the cost $\sum_{i,j \in \Omega} \gamma(i,j) \mathcal{D}(i,j)$ over all possible mappings $\gamma: \Omega^2 \to \mathbb{R}_{\geq 0}$, such that $\sum_{j \in \Omega} \gamma(i,j) = \mu^a(i)$ for all $i \in \Omega$ and $\sum_{i \in \Omega} \gamma(i,j) = \mu^b(j)$ for all $j \in \Omega$.

**Remark 2.17 (Network flow interpretation of EMD).** Equivalently, the EMD is the cost of the minimum-cost admissible flow on the distance network over $\Omega$—the complete, directed graph on $\Omega$ where each edge $(i, j)$ has weight $\mathcal{D}(i, j)$—with supplies $b(\cdot) := \mu^a(\cdot) - \mu^b(\cdot)$. (This interpretation is valid so long as $\mathcal{D}$ is a proper distance metric.)

The generalization of such notions to continuous metric spaces (e.g., Euclidean $\mathbb{R}^d$) requires measure-theoretic considerations resulting in (2.9).

The EMD has a quite general shift-invariance property:

**Proposition 2.18 (Additive invariance of EMD).** Let $\mu^a$, $\mu^b$, and $\bar{\mu}$ be three measures over a finite domain $\Omega$. Then $\mathcal{W}(\mu^a + \bar{\mu}, \mu^b + \bar{\mu}) = \mathcal{W}(\mu^a, \mu^b)$. 

37
Proof. The proof is simply by Remark 2.17 and observing that the supply mapping
\[ b(\cdot) = \mu^i(\cdot) - \mu^b(\cdot) \]
is invariant to the addition. \[ \square \]

Proposition 2.18 formalizes the intuitive notion that adding the same "offset" to two
histograms should not affect the cost of transforming one into the other. Now let the
symbol \( \preceq \) denote a vector inequality, such that in finite domains \( \Omega \), \( \mu' \preceq \mu \)
means that \( \mu'(i) \leq \mu(i) \) for all \( i \in \Omega \). (Such inequality generalizes readily.)

**Corollary 2.19 (Subtractive invariance of EMD).** Let \( \mu^i, \mu^b, \) and \( \bar{\mu} \)
be three measures over a finite domain \( \Omega \), with \( \bar{\mu} \preceq \mu^i \) and \( \bar{\mu} \preceq \mu^b \). Then
\[ W(\mu^i - \bar{\mu}, \mu^b - \bar{\mu}) = W(\mu^i, \mu^b). \]

Proof. The proof is simply by observing that since \( \bar{\mu} \preceq \mu^i \) and \( \bar{\mu} \preceq \mu^b \), then
\[ W(\mu^i, \mu^b) = W((\mu^i - \bar{\mu}) + \bar{\mu}, (\mu^b - \bar{\mu}) + \bar{\mu}). \]
Applying Prop 2.18 obtains the corollary. \[ \square \]

Prop. 2.18 and Corollary 2.19 generalize fully, but the proofs are beyond the scope of
the discussion. The finite-version proofs have been presented for the sake of intuition.

### 2.5.2 Literature Review

One of the most successful recent applications of the EMD has been in image match-
ing and retrieval [127, 99, 35, 101], toward the development of fast computerized
image databases. The EMD obtains several advantages over previously-used metrics
for comparing certain image data represented using histograms (i.e., distributions of
finite support). The metric has also been studied recently from an algorithmic perspec-
tive [64, 77, 9, 106, 8, 65], because classical algorithms to compute the EMD can
be too slow to meet the requirements of large database systems. Many such stud-
ies leverage special structure of a particular ground metric. While most algorithmic
studies of the EMD consider that the two distributions, or histograms, are known \emph{a priori},
a study in [65] considers optimal approximation algorithms in the case that
the distributions are not known, but the samples used to compute the histograms
are obtained as a "streaming input". The EMD has applications in other computer
science domains as well, e.g., alignment of two-dimensional surfaces [78].
Part I
Queueing Models and Control of Fleet Vehicles with Carrying Capacity
Chapter 3

Dynamic Taxiing with Generally Distributed Random Demands in 2- and 3-Dimensions

3.1 Introduction

In this chapter, we study a simple yet insightful model of Mobility-on-Demand systems for one-way trips, where vehicles move by themselves between users. From the modelling perspective, it is irrelevant whether such vehicles are autonomous or piloted by dedicated drivers. We focus our attention on systems with large, homogeneous fleets of vehicles with unit carrying capacity. Our model is based on the one first presented by Swihart and Papastavrou in [119], and likewise draws heavily on the inspiration of the probabilistic models introduced by Bertsimas and van Ryzin [24, 25, 22]. The latter have led to many insightful developments for other fleet-scale robotic planning problems, as discussed in Section 2.3.1.

Specifically, we study a Dynamic Pick-up and Delivery problem (DPDP) in the setting of multiple vehicles with single-integrator dynamics in Euclidean two- and three-dimensional environments. We consider the case that the demands or parcels to be transported arrive randomly over time, with random origin and destination points distributed according to an arbitrary spatial distribution. In this way, our model is significantly more general than its predecessor [119]. Most such studies assume that vehicles have infinite, or unbounded carrying capacity. However, since carrying capacity is a fundamental issue for an agile fleet of inexpensive vehicles, we will focus in this chapter (and throughout the thesis) on the case of limited, in fact, unit carrying capacity; we will refer to this problem as the 1-DPDP.

Contributions. The contributions of this chapter include:

1. a general 1-DPDP model, including multiple vehicles and a general probabilistic model for the generation of origin-destination (O/D) pairs;

2. vehicle routing policies for the light- and heavy-system workload regimes;
3. for policies given, theoretical bounds on the average time customers spend in
the system; compared with
4. theoretical, policy-independent bounds which match or nearly match the policy
bounds from below under certain circumstances;
5. finally, a new checkable condition which is necessary and sufficient to ensure
that stabilizing vehicle routing policies exists.

Discussion. Our main discovery is that, except under a very strict condition on
the demand distribution, and even under the most efficient possible policy, vehicles
will travel empty (without a customer on board) for a positive fraction of the time. In
other words, the average time needed for a vehicle to serve a single customer cannot
be reduced merely to the average trip distance, but is strictly larger by a particular
positive quantity that is fundamental to the system. This phenomenon is due to some
“conservation constraints” observed in the past (see, e.g., see [131]); however, we are
not aware of any previous studies of fundamental bounds.

Structure of the chapter. This chapter is structured as follows. In Section 3.2 we
formally state the problem of the chapter, i.e., we formally define the 1-DPDP we
wish to study. In Section 3.3 we review some existing previous work related to the
DPDP. We provide necessary mathematical preliminaries in Section 3.4. We present
vehicle routing policies for the 1-DPDP in Section 3.5, for each of the light and heavy
demand regimes. In Section 3.6 we derive performance bounds for the given routing
policies, e.g., bounding the average customer waiting times. In Section 3.7 we derive
a new necessary and sufficient condition to ensure that stabilizing policies exist for
the 1-DPDP. We provide policy-independent performance limits in Section 3.8, e.g.,
bounding average customer waiting times from below. We present a simulation study
of policies, performance bounds, and stability in Section 3.9. Finally, in Section 3.10,
we present concluding remarks.

3.2 Problem Statement

In this section, we present the m-vehicle, unit-capacity, Dynamic Pickup and Delivery
problem. For the sake of our pre-occupation with the control-theoretic perspective, we
first define the homogeneous taxi theater in a style similar to that of the homogeneous
repairman theater of Section 2.3 (the setting of the DTRP, i.e., Problem 2.1).

Homogeneous taxi theater Demands arrive into the system according to a ho-

geneous (time-invariant) Poisson process with time intensity \( \lambda \). The \( \text{th} \)
demand requires that a person or parcel be transported by some vehicle from an origin point
\( X_i \in \Omega \) to a destination point \( Y_i \in \Omega \). (\( \Omega \) is the point set.) The origin/destination
pairs \((X_i, Y_i)\) are i.i.d., with joint distribution \( f \). All arrived demands are known
immediately, globally, including their origin and destination information. A homo-
genous fleet of \( m \) vehicles move at maximum speed \( v \) in an environment \((\Omega, \mathcal{F})\). They
have infinite range and servicing quota, but they must remain within a compact,
connected workspace \( \mathcal{W} \subset \Omega \). (Letting \( f_X \) and \( f_Y \) denote the marginal distributions
of the first and second factors of \( f \), respectively, we assume that both \( f_x \) and \( f_y \) are supported entirely on \( W \). Each vehicle has the on-board capacity to carry at most \( Q \in \mathbb{N} \) parcels at a time. Once loaded onto a vehicle, a parcel may be unloaded by the vehicle only at its final destination. The homogeneous taxi theater is specified by the tuple \( ((\Omega, \mathcal{D}), W, f, \lambda, m, v, Q) \). Throughout the thesis, our focus is on the case of unit carrying capacity, i.e., \( Q = 1 \).

Taxi theater events: Various events will occur throughout the execution of a routing policy \( \pi \) (in a taxi theater \( M \)). In this section, we provide the terminology for measuring various spans of time throughout the chapter, either from the perspective of the customer, or from the perspective of the system. Throughout the next discussion, Figure 3-1 provides a valuable illustration for reference.

Consider some demand \( i \). Let \( a_i \) denote the instant that \( i \) arrives, which we call the arrival time. At some later time \( b_i \geq a_i \), a vehicle (say \( j \)) may become free and be assigned to serve \( i \) next. We call \( b_i \) the assignment time. Let \( X_i^- \) denote the location of vehicle \( j \) at time \( b_i \). (Note that very often \( X_i^- \) will be the destination \( Y_{i'} \) of the demand \( i' \) just delivered by \( j \).) Then \( j \) must travel empty for some time \( \sigma_i \geq v^{-1} \mathcal{D}(X_i^-, X_i) \) in order to reach the origin \( X_i \) of \( i \). We call \( \sigma_i \) the fetch time. Let \( c_i \) denote the instant that \( j \) reaches and collects \( i \); we call \( c_i \) the collection time. Finally, vehicle \( j \) carries \( i \) for some time \( \tau_i \geq v^{-1} \mathcal{D}(X_i, Y_i) \) (usually equal) to reach its destination \( Y_i \), deliver it, and become free. We call \( \tau_i \) the carry time, and we denote by \( d_i \) the time that \( i \) is delivered, which we call the delivery time.

Demand \( i \) spends a total time of \( T_i := d_i - a_i \) in the system; we call this the system time. From the perspective of \( i \), it begins to receive service starting from the
collection time $c_i$. We refer to the remaining time $W_i := c_i - a_i$ as the \emph{wait time}. Note, however, that from the system or vehicle perspective the fetch time is also a part of the service of demand $i$, even though $i$ will not necessarily perceive the additional effort. Respecting the system view, we refer to the time $s_i := \sigma_i + \tau_i$ as the \emph{service time}.

Given an homogeneous taxi theater $M$, let $\Pi$ denote the set of all causal, stationary routing policies. A policy is causal if it cannot exploit information about future times. A policy is stationary if its behavior from a given state is the same regardless of time. Given a policy $\pi \in \Pi$, the average system time of demands in steady-state is defined by $T^* = \limsup_{i \to +\infty} E[T_i | \pi]$. We will denote by $\overline{T}^* := \inf_{\pi \in \Pi} \overline{T}_\pi$ the smallest value approachable by policies in $\Pi$.

Now we are in position to state the main problem of the chapter:

\textbf{Problem 3.1 (Dynamic Pickup and Delivery problem).} \\ The $Q$-capacity Dynamic Pickup and Delivery problem ($Q$-DPDP) is to find a policy $\pi^* \in \Pi$ such that the average steady-state system time is as small as possible.

The 1-DPDP can be thought of as a dynamic, multi-vehicle version of the Stacker Crane problem. In the present chapter, we study the problem in the Euclidean spaces, $\Omega = \mathbb{R}^d$, $d \geq 1$; moreover, we focus mainly on $\mathbb{R}^2$ and $\mathbb{R}^3$, which are the most interesting physically spatial dimensions. We will assume that the $m$ vehicles move with negligible turning radius, so that they travel straight-line distances.

\section{3.3 Previous Work}

While there is a fairly extensive coverage in the literature of the stochastic DTRP and its variants (see, e.g., the literature overview in Section 2.3.1), to the author’s best knowledge, the only treatments of the stochastic DPDP are [119, 130, 126, 32]. (We are omitting from that list the works [93, 123, 124, 125] constituent of this thesis.) More prevalent in the literature are \emph{non-probabilistic} treatments of the DPDP, for which thorough surveys on heuristics, meta-heuristics and online algorithms can be found in [20] and [87]. Although the non-probabilistic treatment can be quite effective at solving DPDPs in practice (e.g., in simulation), it often fails to give insights about fundamental limits of performance, e.g., maximum system throughput, customer waiting times, etc. Such insights become critically important as the level of demand scales.

The first and most comprehensive previous work about the DPDP was presented by Swihart and Papastavrou in [119], only several years after the DTRP. Their formulations of the 1-, $k$-, and $\infty$- capacity DPDP were all single-vehicle (single integrator) problems with the fairly restrictive assumption that origin and destination sites are all i.i.d. and uniformly distributed over a square workspace (area $A$).

\textit{Light traffic scenario:} Given such model, Swihart et al. [119] showed that in light traffic, i.e., as $\lambda \to 0^+$, the average time spent by a customer in the system is bounded by

$$\overline{T}^* \geq \nu^{-1} \min_x \mathbb{E}[\|X - x\|] + \overline{\tau},$$

where $\nu = \sqrt{\frac{2}{\pi}}$ is the diffusion constant of the standard Brownian motion.
where $\tau = E_f \|Y - X\|_1/v$. The minimizer of the right-hand expression is called the stochastic median of the origin distribution, and the bound holds regardless of the routing policy. They analyzed several policies in the light traffic scenario, including the Stochastic Queue Median policy, for which they demonstrated that $\lim_{A \to +} T_{\text{SQM}}/T^* = 1$, i.e., the policy is optimal asymptotically.

**Stability conditions:** They also proved a necessary and sufficient condition ($\lambda \tau /v < 1$) to ensure the existence of stabilizing policies, according to Definition 2.16. As the arrival rate $\lambda$ increases, the capacitated and uncapacitated versions of the DPDP adopt quite different properties; thus, we review each case separately:

**Heavy traffic, $\infty$-capacity:** The $\infty$-DPDP, where the on-board capacity of the service vehicle is not bounded, is treated, e.g., in [119, 126, 32]. Swihart et al. [119] introduced and gave system time bounds for several routing policies including the dual TSP policy. Waisanen et al. [126] gave system time bounds under various information constraints; e.g., they considered the case that destination information is revealed only once a demand’s origin is reached. Celik and Modiano [32] developed a model of the DPDP for applications in wireless communication networks, where mobile agents assist in message relay; the on-board capacity of an agent, e.g., in number of bits in memory, is not a practical limitation in such scenario. It turns out (e.g., because of phenomena discussed in [117]), that the $\infty$-DPDP yields to mostly the same analytical techniques as the DTRP.

**Heavy traffic, finite capacity:** The capacity-constrained DPDP, in contrast, does not yield to existing techniques, and thus, there are rather fewer treatments. In fact, to the best of our knowledge, the only analytical study of capacitated DPDP models appears in the original treatment, [119], already discussed, and [130]. In Euclidean $\mathbb{R}^d$, however, it appears that known connections between the SCP and the Bipartite Matching problem can be used (e.g., see [130, Sec. 4.4]) to bring probabilistic results from optimal matching theory to bear.

Swihart et al. gave a policy-independent lower bound on the average system time of the order $(1 - \varrho)^{-2}$, for

$$\varrho = \lambda c_1 \sqrt{A}/v,$$

where $c_1$ is the average distance between two i.i.d. uniformly distributed points in the unit square. ($A$ is the area of the square workspace.) They demonstrated that a Sectoring policy (analogous to well-performing policies for DTRP) has average system time of the order $(1 - \varrho)^{-3}$. Note that, as $\varrho \to 1^-$, the lower bound and the policy bound become arbitrarily far apart. They first introduced the [batched] Stacker Crane policy, based on solving optimally a sequence of static problems (Stacker Crane problems). They could not derive analytical system time bounds; instead, simulation results were used to demonstrate the trend in $\varrho$. However, [130] essentially proved false their conjecture, that the Stacker Crane policy should have average system time of the order $(1 - \varrho)^{-2}$, due to a deep and surprising result given in [5].

Worth mentioning is another work [92], on a closely-related load-rebalancing problem, e.g., of maintaining racks of common-access bicycles placed strategically at locations around a town. (Bicycles are generally a unit-capacity transportation resource.) The authors describe the phenomenon that under general demand distributions in a
discrete environment (transportation is rack-to-rack), bicycles tend to pile up at certain racks, while leaving other racks without any bicycles at all.

3.4 Preliminaries

The analysis of vehicle routing policies in this chapter relies crucially on recent bounds [124, 125] characterizing the lengths of stacker crane tours through large numbers of random O/D pairs. In this section, we introduce several such bounds, reproduced from Chapter 5, where they are derived with formal proof.

Throughout the chapter, we will assume that O/D pairs are i.i.d. in a compact Euclidean workspace \( W \). Let \( f \) denote the distribution of O/D pairs, i.e., \((X_i, Y_i) \sim f\) for all \( i \). Let \( f_X \) denote the distribution of \( X_i \), and let \( f_Y \) denote the distribution of \( Y_i \), the so-called "marginal" distributions of \( f \). Throughout the chapter, we will assume that the following technical conditions hold on \( f \):

1. \( f_X \) and \( f_Y \) are both absolutely continuous, with densities \( \varphi_X \) and \( \varphi_Y \), respectively.
2. \( \varphi_X \) and \( \varphi_Y \) are both \( K \)-Lipschitz, i.e., \(|\varphi(x) - \varphi(y)| \leq K \|x - y\|, \forall x, y \in W\).
3. \( \varphi_X \) and \( \varphi_Y \) are both bounded below and above, i.e., \( 0 < \underline{\varphi} \leq \varphi(x) \leq \overline{\varphi} < \infty \), \( \forall x \in W \).

Throughout the thesis, we will find it useful to express a quantity, defined by

Definition 3.1 (Mover's Complexity).

\[
\mathcal{M}(f) = \mathbb{E}_f \|Y - X\| + \mathcal{W}(f_Y, f_X),
\]

which we refer to henceforth as the Mover's complexity of the demand distribution \( f \). Here, \( \mathcal{W} \) is the Earth Mover's distance introduced in Section 2.5, which for the Euclidean case can be written as

\[
\mathcal{W}(f_Y, f_X) = \inf_{\gamma \in \Gamma(f_Y, f_X)} \int_W \|x - y\| \, d\gamma(y, x).
\]

3.4.1 Lengths of Large (Random) Stacker Crane tours

We begin with an almost sure lower bound on the rate at which stacker crane tours grow, as a function of the number of O/D pairs \( n \):

Theorem 3.2 (Lower bound on SCP* in \( \mathbb{R}^{\geq 1} \)). For any compact, \( d \geq 1 \) dimensional Euclidean workspace \( W \) and any sequence of independent random O/D pairs \( \{(X_k, Y_k)\} \), with distribution \( f \) over \( W^2 \), the length of the optimal stacker crane tour through the first \( n \) O/D pairs (denoted SCP*(\( n; f \))) has limit behavior

\[
\liminf_{n \to +\infty} n^{-1} \text{SCP}^*_f(n) \geq \mathcal{M}(f) \quad \text{almost surely.}
\]
We defer the proof of Theorem 3.2 to Chapter 5. Indeed, we can match the rate of (3.5) with upper bounds that also contain “next-order” terms: In the Euclidean plane \((d = 2)\), we obtain a high-probability upper bound:

**Lemma 3.3 (Upper bound on SCP* in \(\mathbb{R}^2\)).** For any compact, planar workspace \(W \subset \mathbb{R}^2\), there is a constant \(K_W\) such that for any sequence of independent random O/D pairs \(\{(X_k, Y_k)\}\), identically but generally distributed,

\[
\frac{SCP_f(n) - n \mathcal{H}(f)}{\sqrt{n \log n}} \leq K_W
\]

with high probability \((1 - o(1))\) as \(n \to +\infty\).

In higher-dimensional spaces, we obtain almost sure bounds:

**Lemma 3.4 (Upper bound on SCP* in \(\mathbb{R}^{d \geq 3}\)).** For any compact, \(d\)-dimensional Euclidean workspace, \(d \geq 3\), and any sequence of independent random O/D pairs \(\{(X_k, Y_k)\}\), with distribution \(f\) over \(W^2\), there is a constant

\[
\alpha \leq 2 \beta_{M,d} \min_{\varphi \in \{\varphi_V, \varphi_X\}} \left\{ \int_W \varphi(x)^{1/d} \, dx \right\} + \beta_{\text{TSP},d} \int \varphi_X^{1-1/d}(x) \, dx,
\]

such that

\[
\limsup_{n \to +\infty} \frac{SCP_f(n) - n \mathcal{H}(f)}{n^{(d-1)/d}} \leq \alpha.
\]

almost surely. (The constants \(\beta_{\text{TSP},d}\) and \(\beta_{M,d}\) are the matching constant (Section 2.2.2) and the TSP constant (Section 2.2.4), respectively.) If \(X_k\) and \(Y_k\) are independent, i.e., \(Y_k \perp X_k\) (for all \(k\)), then

\[
\alpha \leq \beta_{M,d} \min_{\varphi \in \{\varphi_V, \varphi_X\}} \left\{ \int_W \varphi(x)^{1-1/d} \, dx \right\}.
\]

Lemmas 3.3 and 3.4 are special cases of Theorems 5.18 and 5.17, respectively, which we present with formal proofs in Chapter 5.

**Remark 3.5 (Lower bound on \(mSCP^*\) in \(\mathbb{R}^{d \geq 1}\)).** The \(m\)-vehicle ESCP \((m\text{ESCP})\) consists of finding a partition of the demand set into \(m\) subsets, with a stacker crane tour through each subset such that the total length is minimized. The \(m\text{ESCP}\) might arise, e.g., when \(m\) vehicles are available for service. The bounds of this section hold as stated for the total length of the optimal \(m\) such tours, for any \(m\).

### 3.4.2 Discussion

The bounds of the prequel clearly demonstrate the key role of the Mover’s complexity in the growth of the length of stacker crane tours: Specifically, the average distance traveled between destinations (or between origins), by a single vehicle, approaches such constant in a very strong sense as the number of demands grows large. Of
course, from the famous Law of Large Numbers, we should expect such average to be at least as large as the expected origin-to-destination distance, or average "trip" distance (it makes up the first term of the Mover's complexity). What is remarkable is that it is in fact larger than that by a positive constant which one can characterize precisely: the Earth Mover's distance, which depends only on the workspace geometry and the spatial distribution of demands.

The main technical challenge of this study, which we discuss further in Chapter 5, has been to synthesize sufficiently general probabilistic bounds for the bipartite matching and stacker crane functionals from previously-known bounds which hold only in certain cases. In this chapter, we take these bounds for granted.

These bounds have two important implications for the analysis of policies in the sequel: First, through them, the Mover's complexity (and therefore, the EMD) will appear in stability conditions and performance bounds throughout the thesis, wherever there are capacity constraints. Second, the order of the right-hand sides ("next-order" terms) of (3.6) and (3.8) provide the insight necessary to obtain meaningful performance bounds for simple vehicle routing policies. Such bounds and structural insight provided by the Mover's complexity in particular should provide a system designer with essential information to build business and strategic planning models regarding, e.g., fleet sizing.

### 3.5 Routing Policies

As with most DVRPs, the analysis of the DPDP for all parameter settings is difficult: For the sake of analysis, routing policies should remain simple, yet simple policies tend to perform well only under certain operating conditions. A useful discriminator for the policy design/analysis space is a division into light-load vs. heavy-load regimes, where light-load analysis considers the case that \( \lambda \to 0^+ \), and heavy-load analysis refers to the case that \( \lambda \) grows large. In many cases, there is a threshold rate \( \lambda^* \) (if all other parameters are held fixed), beyond which no policy can stabilize the queue. The fraction \( \rho = \lambda / \lambda^* \) is commonly known as the utilization factor or load factor, and very commonly the system is stabilizable if and only if \( \rho < 1 \). In problems for which \( \lambda^* < +\infty \), "heavy-load" refers to the case that \( \rho \to 1^- \).

In this section we present two vehicle allocation and routing policies for the 1-DPDP, one for each of the two limiting regimes. We will analyse the policies for performance in Section 3.6, in terms of the average system time in steady-state, both for the light-load case (i.e., \( \rho \to 0^+ \)) and for the heavy-load case (i.e., \( \rho \to 1^- \)). In Section 3.8, we derive policy-independent lower bounds for the average steady-state system time, in a manner similar to [22]. We then compare policy performance against these lower bounds.
3.5.1 A policy for light load

The \( m \)-stochastic queue median policy

In this section we briefly describe a policy that achieves asymptotic optimality in the light-load limit. It is essentially a multi-vehicle generalization of the SQM policy of [119]. For an instance of the problem, we consider the placement of \( m \) virtual depots at the "\( m \)-stochastic median" points, i.e., the locations \( x_1^*, \ldots, x_m^* \) within \( W \) such that \( \mathbb{E}_f \left[ \min_{j=1}^m \| X - x_j^* \| \right] \) is minimized. Each depot will correspond to a queue, and is assigned a service vehicle.

The \( m \)-stochastic queue median policy (SQM) — Upon arrival, a demand is assigned to the depot closest to its origin location. The depot’s vehicle serves its demands in first-come first-served order, returning to the depot after each delivery, and waiting there if its queue is empty.

3.5.2 A policy for heavy load

The \( m \)-Stacker Crane policy

In this section we present a simple multi-vehicle generalization of the Stacker Crane policy of [119].

The \( m \)-Stacker Crane policy (SC) — Let \( D \) be the set of outstanding demands waiting for service. If \( D = \emptyset \), the vehicles may move to depot locations (e.g., the \( m \) stochastic median points), where they will wait until \( D \neq \emptyset \). If, instead, \( D \neq \emptyset \), compute the optimal stacker crane tour through all demands in \( D \). Split the tour into fragments in the following way: Follow the tour from a randomly chosen starting point. Record demands in the order that their origins are encountered until the distance traveled first exceeds \( \frac{1}{m} \) of the total tour length (or until the tour is finished). Assign those demands in the given order to some available vehicle. To construct the remaining assignments, repeat the process from where it ended for the previous assignment. Each of the \( m \) vehicles services its share of the demands by following its assigned fragment, and then waits. Repeat the process after all vehicles have completed their fragments.

The Stacker Crane policy is an example of a gated policy: Its execution can be separated into disjoint iterations, where the set of demands which arrive during the \( k \)-th iteration are precisely those which receive service during the \( (k+1) \)-th iteration. The policy is stationary by merit of the fact that the logic is identical in every iteration.

3.6 Performance Analysis

3.6.1 The SQM Policy

The performance of the SQM policy is characterized by the following theorem.
Theorem 3.6 (SQM policy, light load). In light traffic, i.e., as \( \rho \to 0^+ \), with depot locations \( \{x_j\}_j \),

\[
\overline{T}_{SQM} \to v^{-1} \mathbb{E} \left[ \min_{j=1}^m \|X - x_j\| \right] + \tau. \tag{3.10}
\]

**Proof.** The assignment of demands to depots based on their origin location divides the system into \( m \) disjoint M/G/1 queues. Let \( \mathcal{W}_j \) denote the region of dominance of the \( j \)th depot, i.e., \( \mathcal{W}_j := \{x \in \mathcal{W} : \|x - x_j\| \leq \|x - x_j'\| \forall j'\} \). The queue associated with the \( j \)th depot has time intensity \( \lambda_j = \lambda \mathbb{P}[X \in \mathcal{W}_j] \) and a service time distribution with finite variance and mean given by

\[
\bar{s}_j = v^{-1} \mathbb{E} [\|X - x_j\| + \|Y - X\| | X \in \mathcal{W}_j]. \tag{3.11}
\]

Letting \( \bar{T}_j \) denote the average system time of steady-state demands in the \( j \)th queue, and applying the Pollaczek-Khinchin formula for the M/G/1 queue [60], we see that \( \bar{T}_j \to \bar{s}_j \) for all \( j \) as \( \lambda \to 0^+ \). Note that one can write \( \overline{T}_{SQM} = \sum_{j=1}^m \mathbb{P}[X \in \mathcal{W}_j] \bar{T}_j \). Taking the limit as \( \lambda \to 0^+ \) on both sides, and pulling it inside the right-hand sum, obtains (3.10). (Use \( \mathbb{E} [\min_j \|X - x_j\|] = \sum_j \mathbb{P}[X \in \mathcal{W}_j] \mathbb{E} [\|X - x_j\| | X \in \mathcal{W}_j] \).) \( \square 

### 3.6.2 The Stackcrane Policy

In the following discussion, let us define the utilization factor as

\[
\rho := \lambda \mathcal{M}(f)/(mv). \tag{3.12}
\]

The performance of the Stackcrane policy in heavy load is characterized by the following theorem.

**Theorem 3.7 (Stackcrane policy, heavy load).** In \( \mathbb{R}^3 \), the steady-state system time \( \overline{T}_{SC} \) under the Stackcrane policy has limit behavior

\[
\lim_{\rho \to 1^-} \overline{T}_{SC} (1 - \rho)^3 \leq \frac{\alpha^3}{mv \mathcal{M}(f)^2}. \tag{3.13}
\]

In \( \mathbb{R}^2 \),

\[
\lim_{\rho \to 1^-} \overline{T}_{SC} \left[ (1 - \rho)^2 / \log \left( \frac{1}{1 - \rho} \right) \right] \leq \frac{2K_W^2}{mv \mathcal{M}(f)}. \tag{3.14}
\]

(The constants \( \alpha \) and \( K_W \) are from Lemmas (3.4) and (3.3), respectively.)

The proof of Theorem 3.7 builds on an intermediate result, which we will develop shortly. First, however, we will define some notation related to the trajectories of the system:

Recalling that the Stackcrane policy is a gated policy, we refer to the instant \( t_i \) when the \( i \)th stacker crane tour is computed as the \( i \)th epoch. We refer to the time interval between epoch \( i \) and epoch \( i + 1 \) as the \( i \)th iteration of the policy. Let \( n_i \) be the number of demands served during iteration \( i \), and let \( H_i = t_{i+1} - t_i \) be the length of the \( i \)th iteration. The sequences \( \{H_i\} \) of durations and \( \{n_i\} \) of
demands served are closely related. By the chain rule of expectation, one can write
\[ E[n_{i+1}] = E[E[n_{i+1} \mid H_i]]. \]
Since demands arrive according to a Poisson process with rate \( \lambda \), we have \( E[n_{i+1} \mid H_i] = \lambda H_i \). Substituting the right-hand side in the previous expression, we obtain
\[ E[n_{i+1}] = \lambda E[H_i]. \]  
(3.15)

Taking the lim sup on both sides we obtain an asymptotic relation
\[ \bar{n} = \limsup_{i \to +\infty} E[n_i] = \lambda \limsup_{i \to +\infty} E[H_i]. \]  
(3.16)

To prove Theorem 3.7, at the end of the section, we will relate the average system time to the average batch duration and, in turn, to the average batch size \( \bar{n} \) through (3.16). The average batch size is governed by the next result.

**Lemma 3.8 (Stacker Crane policy, batch size, heavy load).** \( \bar{n} \) is finite for all \( \varrho < 1 \), i.e., \( \bar{n} < +\infty \). If \( d = 3 \), then \( \bar{n} \) has the limiting behavior
\[ \limsup_{\varrho \to 1^-} \bar{n}(1 - \varrho)^3 \leq \left( \frac{\alpha}{\mu(f)} \right)^3. \]

If \( d = 2 \), then \( \bar{n} \) has the limiting behavior
\[ \limsup_{\varrho \to 1^-} \bar{n} \left[ (1 - \varrho)^2 / \log \left( \frac{1}{1 - \varrho} \right) \right] \leq 2 \left( \frac{K_W}{\mu(f)} \right)^2. \]

In order to prove Lemma 3.8, let us consider another relation between the iteration duration \( H_i \) and number of demands served \( n_i \). Under the Stacker Crane policy, \( H_i \) is equal to the maximum of the times needed by the \( m \) vehicles to service their assigned fragments of the stacker crane tour, and is bounded surely by
\[ H_i \leq \left( m^{-1} SCP^*(n_i) + C \right) / v, \]  
(3.17)

where \( C \) is a conservative constant, \( C \leq 2 \text{diam} W \), \( \text{diam} W = \max\{\|p - q\| \mid p, q \in W\} \), which accounts for

1. the distance traveled by a vehicle to reach the first origin in its fragment, from its current position;

2. the possible extra mileage required to reach the last destination on the assigned fragment.

Combining (3.17) with (3.15) obtains a recurrence relation
\[ E[n_{i+1}] \leq \lambda \left( m^{-1} E[SCP^*(n_i)] + C \right) / v. \]  
(3.18)

Since both forms of \( SCP^* \), i.e., (3.6) for \( d = 2 \) and (3.8) for \( d \geq 3 \), are concave functions, the consequences of ignoring any lower order terms are not significant to
the analysis we may apply Jensen’s inequality for concave functions to obtain

$$
E[n_{i+1}] \leq \lambda \left( m^{-1} \text{SCP}^*(E[n_i]) + C \right) /v.
$$

(3.19)

Lemma 3.8 is ultimately proved by a straightforward application of the next two system bounds to this recurrence relation.

**Lemma 3.9.** Consider a non-negative, scalar system obeying

$$
y(i + 1) \leq a y(i) + b y(i)\gamma + e(y(i)),
$$

(3.20)

with \(a, b \geq 0, \gamma \in (0, 1)\), and a non-negative, continuous function \(e(\cdot)\) bounding low-order terms; i.e., \(e(y) = o(y^\gamma)\), i.e., \(\lim_{y \to +\infty} e(y)/y^\gamma = 0\). If \(a < 1\), then from any initial condition \(y(1) \geq 0\), \(\sup_{i \to +\infty} y(i) < +\infty\); that is, the system trajectories remain bounded. Moreover,

$$
\lim_{a \to 1^-} \sup_{i \to +\infty} (1-a)^{1/(1-\gamma)} \lim_{i \to +\infty} y(i) \leq b^{1/(1-\gamma)}.
$$

**Lemma 3.10.** Consider a non-negative, scalar system obeying

$$
y(i + 1) \leq a y(i) + b \sqrt{y(i) \log y(i)} + e(y(i)),
$$

(3.21)

with \(a, b \geq 0\), and a continuous function \(e(\cdot)\) bounding low order terms. If \(a < 1\), then from any initial condition \(y(1) \geq 0\), \(\sup_{i \to +\infty} y(i) < +\infty\), i.e., the system trajectories remain bounded. Moreover,

$$
\lim_{a \to 1^-} \sup_{i \to +\infty} \left[ (1-a)^2 / \log \left( \frac{1}{1-a} \right) \right] \lim_{i \to +\infty} y(i) \leq 2b^2.
$$

For the technical proofs of these Lemmas, see Appendix A.

Finally, applying these lemmas, we can prove Lemma 3.8 and in turn Theorem 3.7.

**Proof of Lemma 3.8.** The proof for the case \(d = 3\) is by combining (3.19) with (3.8) and applying Lemma 3.9. The proof of Lemma 3.8 for the case \(d = 2\) is by combining (3.19) with (3.6) and applying Lemma 3.10. \(\Box\)

**Proof of Theorem 3.7.** Since a steady-state demand \(i\) is equally likely to arrive at any moment during its “arrival” iteration, and since it is equally likely to be served in any order during the next iteration (the order of the demands within a stacker crane tour is not determined by their temporal data), we can bound the expected system time \(E[T_i]\) of a steady-state demand \(i\), by

$$
E[T_i] \leq \frac{1}{2} E\left[H_{k(i)}\right] + \frac{1}{2} E\left[H_{k(i)+1}\right],
$$

where \(k(i)\) is the iteration during which \(i\) arrives. Since the number of served batches
goes to $+\infty$ as $i \to +\infty$,
\[
\bar{T}_{\text{sc}} \leq \limsup_{i \to +\infty} \mathbb{E}[T_i] \leq \limsup_{k \to +\infty} \frac{1}{2} \mathbb{E}[H_k] + \frac{1}{2} \mathbb{E}[H_{k+1}]
\]
\[
= \limsup_{k \to +\infty} \mathbb{E}[H_k] = \bar{n}/\lambda;
\]
the last equality follows from (3.16). The rest of the proof is by substituting $\lambda = gmv/\mathcal{M}(f)$ in the inequality, and applying Lemma 3.8. 

A distributed Stacker Crane policy?

We studied also another version of the Stacker Crane policy, whereby the demand arrival process would be split evenly into $m$ identical subprocesses, one for each vehicle (e.g., one could assign demands to vehicles in a round-robin manner, or choose each demand assignment at random uniformly among vehicles, etc.) Each vehicle then would run the single-vehicle Stacker Crane policy on its own subprocess, ignoring all other demands. Such “process splitting” leads to heavy-load scaling of the system time, with respect to the number of vehicles, a factor $m$ greater than that of the centralized Stacker Crane policy. Hence, as a general rule, it appears that process splitting should be avoided in one-way vehicle sharing systems for large fleets.

3.7 Stability Condition for 1-DPDP

In the previous section, we showed that the Stacker Crane policy stabilizes the 1-DPDP as long as $g < 1$, for $g$ defined by (3.12):
\[
g := \lambda \mathcal{M}(f)/(mv).
\]

In this section we show that $g < 1$ is indeed necessary for the 1-DPDP to admit any stabilizing policy; thus, it is both a necessary and sufficient condition. In the sense that the Stacker Crane tour is stabilizing whenever stability is possible, we call it a perfectly stabilizing policy.

Let us recall the control-theoretic notion of stability from Section 2.3:

**Definition 2.16 (Stabilizable Theater).** Given a theater $\mathcal{M}$, a routing policy $\pi$ is said to be stabilizing if $\bar{T}_\pi < +\infty$, i.e., the average time spent in the system by demands remains uniformly bounded for all times. Then the system $(\mathcal{M}, \pi)$ is said to be stable. If the set of policies $\{ \pi \in \Pi | \bar{T}_\pi < +\infty \}$ is not empty, i.e., $\bar{T} < +\infty$, then $\mathcal{M}$ is stabilizable.

**Theorem 3.11 (Stability condition for 1-DPDP).** In any $d \geq 1$ dimensional Euclidean workspace $W$, the unit-capacity homogeneous taxi theater (setting of the 1-DPDP) is stabilizable [in the sense of Definition 2.16] if and only if $g < 1$. 

53
Remark 3.12. When \( f_Y = f_X \), then one has \( \mathcal{W}(f_Y, f_X) = 0 \) (by Proposition 5.9), and (3.12) and Theorem 3.11 reduce to the versions in previous works [119, 130, 93] under more restrictive assumptions.

Proof of Theorem 3.11—Part I: Sufficiency. A constructive proof of sufficiency was provided by the analysis of the Stacker Crane policy, e.g., Theorem 3.7, which proved that it is stabilizing for all \( \varrho < 1 \). \( \square \)

Proof of Theorem 3.11—Part II: Necessity. Consider any causal, stable routing policy (since the policy is stable and the arrival process is Poisson, the system has renewals and the inter-renewal intervals are finite with probability one). Let \( A(t) \) be the number of demand arrivals from time 0 (when the first arrival occurs) to time \( t \). Let \( R(t) \) be the number of demands in the process of receiving service at time \( t \) (a demand is in the process of receiving service if a vehicle has been assigned to it and it has not been delivered yet). Recall that \( s_i \) is the total service time (both fetch and carry) of demand \( i \).

The time-average number of demands receiving service is given by

\[
\bar{q} := \lim_{t \to +\infty} \frac{1}{t} \int_0^t R(\tau) d\tau.
\]

By following the arguments in [60, p. 81-85, Little’s Theorem], \( \bar{q} \) can be written as:

\[
\bar{q} = \lim_{t \to +\infty} \frac{\sum_{i=1}^{A(t)} s_i}{t} = \lim_{t \to +\infty} \frac{\sum_{i=1}^{A(t)} s_i}{A(t)} \lim_{t \to +\infty} \frac{A(t)}{t},
\]

where the first equality holds almost surely and all limits exist almost surely. The second limit on the right is, by definition, the arrival rate \( \lambda \). The first limit on the right can be lower bounded as follows:

\[
\lim_{t \to +\infty} \frac{\sum_{i=1}^{A(t)} s_i}{A(t)} \geq \lim_{t \to +\infty} \frac{1}{v} \frac{m-SCP^*(A(t))}{A(t)},
\]

where \( m-SCP^*(A(t)) \) is the total length of the optimal \( m \)-vehicle stacker crane tour through the \( A(t) \) demands (i.e., it is the optimal solution to the multi-vehicle ESCP—see Remark 3.5). Since \( A(t) \to +\infty \) almost surely as \( t \to +\infty \), by Theorem 3.2 and Remark 3.5 we can write, almost surely,

\[
\lim_{t \to +\infty} \frac{m-SCP^*(A(t))}{A(t)} \geq \mathcal{M}(f).
\]

Collecting the above results, we obtain \( \bar{q} \geq \frac{\lambda \cdot \mathcal{M}(f)}{v} \), almost surely. Since the arrivals are non-deterministic, stability requires that the average number of demands receiving service must be strictly less than the number of servers, i.e., \( \bar{q}/m < 1 \); this implies that for any causal, stable routing policy \( \lambda \cdot \mathcal{M}(f)/(mv) < 1 \), proving the necessary condition. \( \square \)
The stability condition in Theorem 3.11 depends on the workspace geometry, the distributions of pickup and delivery points, the demands' arrival rate, and the number of vehicles, and makes explicit the roles of the different parameters in affecting the stability of the overall system.

### 3.8 Performance Lower Bounds

In this section we present two lower bounds: the first one is most useful as $\rho \to 0^+$ (light load), while the second one holds as $\rho \to 1^-$ (heavy load). The heavy load lower bound holds only in the case that $f_Y = f_X$. It is an open question to produce a heavy load lower bound for the case $f_Y \neq f_X$; it cannot be obtained by the common techniques.

#### 3.8.1 A light load lower bound

A lower bound that is most useful in light load (i.e., when $\rho \to 0^+$) is the following.

**Theorem 3.13 (System time, light load bound).** The expected time spent in the system by a demand is bounded below by

$$
T^* \geq v^{-1} \min_{(x_1, \ldots, x_m) \in W^m} \mathbb{E}_{f_X} \left[ \min_{j=1}^{m} \|X - x_j\| \right] + \overline{\tau}.
$$

*(3.22)*

**Proof.** The proof is rather straightforward, following logic similar to that in [119]. The system time of a demand is composed of the wait time followed by the carry time. The average carry time is simply proportional to the average origin-to-destination distance, i.e., $\overline{\tau} = \mathbb{E}_f \|Y - X\|/v$. If the vehicles are in positions $x_1, \ldots, x_m$ when the demand arrives, then the wait time is bounded below by the time to reach the origin by the closest vehicle, i.e., $W \geq \min_{j=1}^{m} \|X - x_j\|/v$. Supposing that we could place the vehicles in the a-priori best positions (i.e., respecting causality), then the wait time is bounded in expectation by $\overline{W}^* \geq \min_{(x_1, \ldots, x_m) \in W^m} \mathbb{E}_{f_X} \left[ \min_{j=1}^{m} \|X - x_j\| \right]$. Adding the two terms obtains (3.22).

Comparing this result with Theorem 3.6, using depots at the $m$ stochastic median points, shows the tightness of the lower bound and the optimality of the SQM policy in the light-load limit.

#### 3.8.2 A heavy load lower bound

In this section we present system time lower bounds that holds in the heavy-load limit (i.e., $\rho \to 1^-$). Unfortunately, the bound holds only when $f_Y = f_X$. To derive this bound we make heavy usage of the proof techniques developed in [22]. The two bounds of this section make a distinction between policies that satisfy a certain kind of “fairness” condition and those which do not. We start with a technical definition of the fairness condition.
Definition 3.14 (Origin-fair policies). Given any set $A \subseteq \mathcal{W}$, let $\bar{W}_X(A)$ denote the average waiting time in steady-state of demands whose origins lie in $A$. A policy $\pi$ is said to be origin-fair if $\bar{W}_X(A_1) = \bar{W}_X(A_2)$ for every pair of sets $A_1, A_2 \subseteq \mathcal{W}$. Note that this implies $\bar{W}_X(A) = \bar{W}_X(W)$ for every set $A \subseteq \mathcal{W}$.

For example, the Stacker Crane policy is an origin-fair policy. Because the stacker crane tour is started from a random location, the waiting time of a demand within any iteration has no conditioning on any of its spatial data.

Theorem 3.15 (System time, heavy load bound). Within the class of origin-fair policies in Euclidean $\mathbb{R}^d$, $d \in \{2, 3\}$, with $\varphi_Y = \varphi_X = \varphi$,

$$\lim_{e \to 1^-} T^e (1 - q)^d \geq \frac{\gamma_d}{m u \bar{W}^{d-1}} \left[ \int_{\mathcal{W}} \varphi(x)^{(d-1)/d} \, dx \right]^d. \quad (3.23)$$

where $\gamma_2 = \frac{2}{3 \sqrt{\pi}}$ and $\gamma_3 = \frac{(3/4)^{4/3}}{\sqrt[4]{\pi}}$.

Before proving Theorem 3.15, we must develop one intermediate result. According to Little’s law (see [60]), the average number of outstanding demands with origin sites in an arbitrary area $A \subseteq \mathcal{W}$ can be expressed as $N_X(A) = \lambda_X(A) \bar{W}_X(A)$, where $\lambda_X(A) = \lambda \int_A \varphi_X(x) \, dx$ and $\bar{W}_X(A)$ is the average waiting time of a demand whose origin lies in $A$. We only consider origin-unbiased policies, and so $\bar{W}_X(A) = \bar{W}$. Since Little’s law also holds that $N = \lambda \bar{W}$, one can combine the previous results to obtain

$$\bar{N}_X(A) = \bar{N} \int_A \varphi_X(x) \, dx. \quad (3.24)$$

Because of equation (3.24), and because $\varphi_X$ is Lipschitz, given a ball $B(x, z) = \{x' \in \mathcal{W} \mid \|x' - x\| \leq z\}$ one can write

$$\bar{N}_X(B(x, z)) = \bar{N} [\varphi_X(x) V_d z^d + o(z^d)]. \quad (3.25)$$

where $V_d$ is the volume of a unit ball in $\mathbb{R}^d$, e.g., $V_2 = \pi$ and $V_3 = 4\pi/3$. Here, the “little-o” notation indicates $\lim_{z \to 0^+} o(f(z))/f(z) = 0$.

Lemma 3.16 (Distance to closest origin). Let $Z$ be the distance in steady-state from a vehicle at some delivery epoch to the nearest origin of an outstanding demand. For any origin-fair policy in $\Pi$

$$\lim_{N \to +\infty} \bar{N}^{1/d} \mathbb{E} [Z] \geq \gamma_d \int_{\mathcal{W}} \varphi^{(d-1)/d}(x) \, dx, \quad (3.26)$$

where $\gamma_d = \frac{d}{d+1} V_d^{-1/d}$.

Proof. We first condition on the site (destination) of the recent delivery, $Y' = y$, and write

$$\mathbb{P}[Z \leq z | Y' = y] = \mathbb{P}[N^+_X(B(y, z)) > 0] \leq \bar{N}^+_X(B(y, z)).$$
where \( N^+_x \) is the number of outstanding demands with origins in the ball \( B(y, z) \) at the delivery epoch; \( \overline{N}^+_x \), following the usual convention, is its expectation in steady-state. The inequality holds because \( N^+_x \) is a non-negative, integer-valued random variable. For Poisson arrival processes, it holds that \( \overline{N}^+_x(A) = \overline{N}^+_x(A) \) (this is a consequence of the PASTA property, see [129]), and recalling equation (3.25) we obtain

\[
\overline{N}^+_x(B(y, z)) = \overline{N} \left[ \varphi_X(y) V_d z^d + o(z^d) \right].
\]

Using the identities \( \mathbb{E}[Z] = \int_0^{\infty} \mathbb{P}[Z > z] \, dz \) (for non-negative random variables) and \( \mathbb{P}[Z \leq z] + \mathbb{P}[Z > z] = 1 \), we can write

\[
\mathbb{E}[Z \mid Y' = y] = \int_0^{\infty} 1 - \mathbb{P}[Z \leq z \mid Y' = y] \, dz \\
\geq \int_0^{\infty} 1 - \overline{N} \left[ \varphi_X(y) V_d z^d + o(z^d) \right] \, dz.
\]

Note that we can restrict the upper limit of the integral arbitrarily without consequence, since it is a lower bound and the integrand is non-negative. Letting \( c(y) := \overline{N} \varphi_X(y) V_d \), and limiting the integral to \( \overline{c}^{-1/d}(y) \), we obtain

\[
\mathbb{E}[Z \mid Y' = y] \geq \int_0^{\overline{c}^{-1/d}(y)} 1 - c(y) z^d - \overline{N} o(z^d) \\
= \overline{N}^{-1/d} \frac{d}{d+1} V_d^{1/d} \varphi_X(y)^{-1/d} - o(\overline{N}^{-1/d}).
\]

We may remove the conditioning on \( Y' \) as follows, obtaining

\[
\mathbb{E}[Z] = \int \mathbb{E}[Z \mid Y' = y] \varphi_Y(y) \, dy \\
\geq \overline{N}^{-1/d} \frac{d}{d+1} V_d^{1/d} \left[ \int \varphi_X(y)^{-1/d} \varphi_Y(y) \, dy \right] - o(\overline{N}^{-1/d}).
\]

We obtain the lemma by applying the condition that \( \varphi_X = \varphi_Y = \varphi \), multiplying by \( \overline{N}^{1/d} \) on both sides, and taking the limit as \( \overline{N} \to +\infty \).

We are now in a position to prove the main result of this section.

**Proof of Theorem 3.15.** To ensure that \( \overline{N} < +\infty \), it must hold that

\[
\lambda s/m = \frac{\lambda}{m} [\mathbb{E}[\sigma] + \mathbb{E}_f \|Y - X\|/v] = \frac{\lambda}{m} \mathbb{E}[\sigma] + \varrho \leq 1.
\]

(3.27)

In the heavy-traffic limit, as \( \overline{N} \to +\infty \), the distance \( Z \) becomes a lower bound for the fetch distance \( \langle \sigma \rangle \) with arbitrarily high probability. Therefore, multiplying the second-to-last term of (3.27) by \( v/v \), rearranging, and substituting \( \mathbb{E}[Z] \leq v \mathbb{E}[\sigma] \), we
obtain
\[ \frac{\lambda}{m v} E[Z] \leq 1 - \varrho. \] (3.28)

Multiplying both sides by \( N^{1/d} \) and raising to the power \( d \), we obtain
\[ \overline{N}(1 - \varrho)^d \geq \frac{\lambda^d [N^{1/d} E[Z]]^d}{(m v)^d}. \]

Applying Little’s Law, i.e. \( \overline{N} = \lambda \overline{W} \), and substituting \( \lambda = \varrho m v / \tau \), we get
\[ \overline{T}(1 - \varrho)^d \geq \overline{W}(1 - \varrho)^d \geq \frac{\varrho^{d-1}}{m v \tau^{d-1}} (N^{1/d} E[Z])^d. \]

Finally, in the limit as \( \varrho \to 1^- \), (3.28) implies that \( E[Z] \to 0^+ \). For the proximity of closest points to vanish in this way, it requires that \( \overline{N} \to +\infty \). Applying Lemma 3.16 we obtain the claim. \( \square \)

Spatially biased policies, by enlargement of the set of policies, should potentially outperform spatially unbiased policies. We extend a Theorem from [22] to see how much performance might be improved:

**Proposition 3.17.** For causal and stationary, but potentially spatially unfair policies, the system time lower bound is
\[ \lim_{\varrho \to 1^-} \overline{T}(1 - \varrho)^d \geq \frac{\gamma_d^d}{m v \tau^{d-1}} \left[ \int_{W} \varphi(x)^{d/(d+1)} \, dx \right]^{d+1}. \] (3.29)

The main insight to prove Theorem 3.15 was to obtain a density function \( \phi \) for \( \overline{N}_X \), i.e., \( \phi \) such that \( \overline{N}_X(A) = \overline{N} \int_A \phi(x) \, dx \) for all \( A \subseteq \mathcal{W} \). For origin-fair policies, we argued using Little’s law that \( \phi = \varphi_X \). For biased policies, instead, we will re-trace the previous logic with \( \phi \) unknown (i.e., variable), and then minimize the resulting expression over valid densities, i.e., \( \phi(x) \geq 0 \) everywhere and \( \int_W \phi(x) \, dx = 1 \).

**Proof.** Substituting \( \phi \) instead of \( \varphi_X \) in (3.24), and repeating the proof of Lemma 3.16, one obtains
\[ \lim_{N \to +\infty} \overline{N}^{1/d} E[Z] \geq \gamma_d \int_W \phi^{-1/d}(x) \varphi_X(x) \, dx. \] (3.30)

Using the Lagrangian relaxation and variational arguments in [22, p.12], the global minimizer of the right-hand side over valid density functions is
\[ \phi^*(x) := \left[ \int_W \varphi_X(x)^{d/(d+1)} \, dx \right]^{-1} \varphi_X(x)^{d/(d+1)}. \]

Substituting \( \phi^* \) into (3.26) obtains
\[ \lim_{N \to +\infty} \overline{N}^{1/d} E[Z] \geq \gamma_d \left[ \int_W \varphi_X^{d/(d+1)}(x) \, dx \right]^{(d+1)/d}. \]
Using this expression instead of (3.26) in the proof of Theorem 3.15 obtains the result.

3.8.3 Discussion

Although the heavy-load lower bounds of the section can only be guaranteed under the special condition that \( f_Y = f_X \), comparing them to the system time bounds obtained by the Stacker Crane policy offers some assurance about the performance of that policy. Theorem 3.7 is strongest in \( \mathbb{R}^3 \), and when origin and destination points are independently distributed, i.e., \( Y \perp X \). Then it can be shown by combining Theorems 3.7 and 3.15 with \( d = 3 \), that the average system time under the Stacker Crane policy differs from the lower bound for any origin-fair policy by at most the constant factor \( (\beta_{M,3}/\gamma_3)^3 \). Thus, the system time under the Stacker Crane policy has the same growth rate in terms of \( \varrho \) as the policy-independent lower bound. In this sense we say the the Stacker Crane policy is efficient. Even if the origins and destinations are not independent, i.e., \( Y \not\perp X \), this constant increases at most to \( \gamma_3^{-3}(2\beta_{M,3} + \beta_{TSP,3})^3 \). In either case the constant factor is universal, i.e., it is independent of the number of vehicles, the arrival rates of demands, and the spatial density for the demand locations, etc. Among all causal and stationary policies, including potentially biased ones, the Stacker Crane policy remains efficient. However, the gap between the upper and lower bounds will include an additional constant factor \( \left( \int_Y \varphi(x)^{2/3} \, dx \right)^3 / \left( \int_Y \varphi(x)^{3/4} \, dx \right)^4 \). While for any density function \( \varphi \) that gap is a constant, the constant could be arbitrarily large by choice of \( \varphi \) (as argued in [22, p.24]). We believe that indeed the upper bound is not tight in this case, and the average system time might be reduced at the cost of allowing policies with spatial bias.

It remains an open question whether the Stacker Crane policy is efficient in the two-dimensional case (even when \( f_Y = f_X \)). However, comparing Theorems 3.7 and 3.15 with \( d = 2 \), we can see it is not inefficient by a factor greater than \( \log_{1/\varrho} \).

It remains an open question to determine lower bounds when \( f_Y \neq f_X \). (Unfortunately, the prevalence of trips of non-vanishing length in that case disqualifies the standard arguments based on nearest neighbor proximity.)

3.9 Simulation in Heavy-load

In this section, we present simulation results to support the theoretical findings of the chapter, focusing mainly of the heavy-load scenario. In particular, our simulation experiments examine the two main contributions of the chapter: performance bounds for the Stacker Crane policy, and stability conditions for the 1-DPDP.

We consider several scenarios for the generation of demands, described below. In every case the origins and destinations are independent, i.e., \( Y \perp X \):

Case I Uniform over \([0,1]^3\): In the first scenario, origins and destinations alike are uniformly distributed over the unit cube.
Figure 3-2: Illustration of non-uniform distributions (Cases III and IV), by sampling: $n = 100$ samples for each distribution; origin sites are shown as (red) triangle markers; destination sites are shown as (blue) circles.

**Case II** Uniform over $[0, 1]^2$: In the second scenario, origins and destinations alike are uniformly distributed over the unit square.

**Case III** Unit Cubes Arrangement: In the third scenario, the distribution of origin sites $f_X$ places one-half of its probability uniformly over a unit cube centered along the $x$-axis at $x = -4$, and the other half uniformly over the unit cube centered at $x = -2$. The distribution of destination sites $f_Y$ places one-half of its probability uniformly over the cube at $x = -4$ and the other half over a new unit cube centered at $x = 2$.

**Case IV** Co-centric Spheres Arrangement: In the fourth scenario, origins are uniformly distributed over a sphere of radius $R = 2$, and destinations are uniformly distributed over a sphere of radius $r = 1$. Both spheres are centered at the origin.

For the purpose of illustration, Figures 3-2(a) and 3-2(b) show $n = 100$ points sampled from each of $f_X$ and $f_Y$, in Cases III and IV, respectively.

The cases under consideration are special examples for which one can compute the Earth Mover's distance of (3.4) explicitly (exploiting geometrical symmetry), and also the right-hand side of (3.7). The expectation $\bar{\tau} = \mathbb{E}_f \|Y - X\|$ can be approximated quite closely, e.g., by stochastic averaging, to complete the expression (3.3) of the Mover's complexity $\mathcal{M}(f)$. All such values have been computed numerically, and compiled in Table 3.1.$^1$

$^1$Recall that $\beta_{3,3} \approx 0.708$. 

60
Case I

\[ 6.62 \times 10^{-1} \]

\[ 8.10 \times 10^{-1} \]

Case II

\[ 5.21 \times 10^{-1} \]

N/A

N/A

Case III

\[ 3.20 \]

\[ 5.20 \]

\[ 2.62 \times 10^{-2} \]

Case IV

\[ 1.66 \]

\[ 0.75 \]

\[ 2.41 \]

\[ 2.59 \times 10^{-1} \]

Table 3.1: Compiled statistics for simulated demand models, with estimates of \( E_f \|Y - X\|, \mathcal{W}, \mathcal{M}, \) and \( \alpha. \)

3.9.1 Performance

First, we present results from simulations of the Stacker Crane (SC) policy for the 1-DPDP in heavy-load conditions. We provide simulation results for all four demand generation models. Simulations were run until the per-demand system time (a running average) met a strict empirical convergence condition; roughly speaking, until the average failed to deviate significantly from a flat trajectory for a long time. To execute the Stacker Crane policy, stacker crane tours were computed using a heuristic presented in [130, Sec. 4.4]. The heuristic is quite similar to the LARGEARCS algorithm of [57], and we will discuss all such algorithms in great detail in Chapter 5. The heuristic depends on subroutines to compute optimal Euclidean bipartite matchings and TSP tours; a Python implementation of the Kuhn-Munkres assignment algorithm [82] was used to generate optimal matchings; approximately optimal TSP tours were computed using linkern² software.

Case I: The plot in Figure 3-3 demonstrates for Case I the dependence of the average system time, empirically, on the load factor \( g. \) Each point represents a single 1-DPDP simulation, with a single, unit-speed vehicle, run for sufficiently long that the per-demand average system time "stabilized". Since all other parameters in (3.12) are fixed, target utilizations were realized by setting the rate \( \lambda \) appropriately. The \( x \)-axis of the plot is derived from the left-hand side of (3.13), with the goal of obtaining [convergence to] a linear trend. The top axis is annotated with the corresponding utilization values, in a range of \( \approx 0.5 - 0.7 \) i.e., a moderate-to-heavy workload. Unfortunately, computing SCP tours proved to be extremely time consuming for batches of size greater than about 200 demands, and such batches occurred regularly in simulations of any utilization \( g > 0.75. \) Fortunately, the average system time in the simulated regime indeed is quite well predicted by the upper bound of (3.13).

We also studied how the fleet size affects system time. The plots in Figures 3-4(a) and 3-4(b) show results of simulations of Case I, with \( m = 2 \) and \( m = 5 \) vehicles, respectively. (The simulations are over the same range of utilization factors.) Several new observations can be made: First, the resulting points do not lie beneath the bound line. Second, they belie the curvature of a sub-linear trend for lower system workloads. Figure 3-4(a) shows a dashed trend line through points in the latter half of the data, for \( m = 2 \) vehicles. Its slope is close to that of the bound line, suggesting

---

² The TSP solver linkern is freely available for academic research use at http://www.tsp.gatech.edu/concorde.html.
that the sub-linear term indeed yields to the predicted linear bound after a transient. (The theory only requires that the slope of that trend should become less than that of the bound line shown.) In Figure 3-4(b), for $m = 5$ vehicles, the alleged transient does not yield within the range of workloads simulated, however, one could easily suspect it would if heavier utilization could be simulated. Thus, the transient appears to be longer for larger fleet sizes. Unfortunately, it has proved quite challenging to simulate heavier workload regimes in $\mathbb{R}^3$.

Figure 3-5 shows plots of the average system time under the SC policy for a fixed utilization factor $\rho = 0.678$, and with fleet size $m$ in the range 1–10. Note that by fixing $\rho$, we are letting the lower bound in Theorem 3.15 and the upper bound in Theorem 3.7 scale as $1/m$. The results indeed show a decreasing trend, which, however, appears to be weaker than the theoretical $1/m$ trend; this is perhaps due to one or a combination of factors: First, we are using approximations for optimal SCP and TSP solutions. Second, and most likely, the regime of $\rho = 0.678$ may not be sufficiently heavy to exhibit asymptotic behavior for a large number of vehicles. Note, for example, that the bounds on the number of outstanding demands in Lemma 3.8 do not depend on the number of vehicles. Thus, splitting the original stacker crane tour among large numbers of vehicles (in each iteration of the SC policy) may result in fragments which are short relative to the extra distances required to reach them. Unfortunately, it has been computationally impractical to simulate scenarios with higher utilization, because of the runtime complexity of existing heuristics.

**Case II:** The plot in Figure 3-6 shows the average system time versus utilization factor for 1-DPDP simulations of Case II, with a single, unit-speed vehicle. The $x$-axis
Figure 3-4: Average system time versus workload, in simulation of Case I, under Stacker Crane policy, with \( m = 2 \) and \( m = 5 \) unit-speed vehicles.

Figure 3-5: Average system time versus fleet size, in simulation of Case I with fixed utilization \( \rho = 0.678 \), under Stacker Crane policy.

of this plot is derived from the left-hand side of (3.14), instead of (3.13), because the workspace is planar. In this case, the average system time is quite well predicted by a linear trend, although in \( \mathbb{R}^2 \) we do not have any explicit constant of proportionality to obtain a bound. The dashed line shows a linear regression of the data points.

**Case III:** The plot in Figure 3-7 shows the average system time versus utilization factor for 1-DPDP simulations of Case III, with a single, unit-speed vehicle. The \( x \)-axis of the plot is once again derived from the left-hand side of (3.13). Case III has the largest Mover’s Complexity of all the cases under consideration, which seems to have some interesting consequences. The main *analytical* result is that \( \alpha \) is quite small, and so the system time should be fairly insensitive to the \( x \)-axis statistic, \((1 - \rho)^{-3}\). Indeed, this property can be verified in the plot. Such property can be quite deceiving, however, since the span of the \( x \)-axis actually covers a very small range of utilization
\[ \rho = \lambda M / (mv) \]

Figure 3-6: Average system time versus workload in Case II, under Stacker Crane policy with one unit-speed vehicle.

...factors, i.e., \(0.8 \leq \rho \leq 0.9\), shown on the top axis. Therefore—and, moreover, since \(M\) is relatively large—the system time is actually *very* sensitive to small changes in the demand arrival rate. Another interesting feature of Case III is a large transient, which spans values of \((1 - \rho)^{-3}\) beyond those for which Case I could be simulated effectively.

Although the average system time seems to approach a linear trend after a small transient, the slope of that trend (the dashed line through the data) does not appear to be as small as predicted. One possible explanation is that we may have some error in the value computed for \(M\). Several dashed curves are shown on the plot, demonstrating how the upper bound in terms of another hypothetical \(\tilde{M}\) would be warped by our plot according to the computed estimate \(\tilde{M}\) (the value from Table 3.1). Each curve is annotated with the associated ratio \(\tilde{M} / M\). Note that in the simulated regime, the observed data could be explained with between 1-3% hypothetical error in our estimate of the Mover’s complexity. This observation demonstrates that the probabilistic analysis of MoD systems can be extremely sensitivity to the precision of the Mover’s complexity.

*Case IV:* The plot in Figure 3-8 shows the average system time versus utilization factor for 1-DPDP simulations of Case IV, with a single, unit-speed vehicle. The dashed blue dashed line is a trend fit through the data, and the green dashed line is the upper bound line. The linear trend is fairly strong after a sharp transient, and the slope of the trend is, for the first time observed so far, *strictly* less than the slope of the bound line.
\[ \rho = \lambda \mathcal{M} / (mv) \]

Figure 3-7: Average system time versus workload in Case III, under Stacker Crane policy with one unit-speed vehicle.

\[ (1 - \rho)^{-3} \]

\[ \rho = \lambda \mathcal{M} / (mv) \]

Figure 3-8: Average system time versus workload in Case IV, under Stacker Crane policy with one unit-speed vehicle.
3.9.2 Stability Conditions

We conclude the simulation section with some heuristic validation of Theorem 3.11 and the resulting threshold $\lambda^* := \frac{mv}{\mathbb{M}}$ separating stabilizable arrival rates from unstabilizable ones. We consider only Cases III and IV, above, for which $\mathbb{M} \neq 0$.

Experiment Design: The main insight of this section is as follows. Let $\pi$ be a policy for the 1-DPDP that is perfectly stabilizing, i.e., stabilizing for all $\lambda < \lambda^*$. (The policy should also satisfy some technical “fairness” conditions, which gated policies satisfy automatically.) We consider the 1-DPDP under $\pi$ with $\lambda > \lambda^*$. Clearly, since $\lambda > \lambda^*$, the number of outstanding demands in the system grows unbounded. However, we should expect the policy to serve demands at an average rate approaching $\lambda^*$ (the fastest rate under $\pi$). Thus, the number of outstanding demands should grow at an average rate of $\lambda - \lambda^*$. Since we can control $\lambda$ in simulation, we can use this insight to estimate $\lambda^*$, e.g., by the simple calculation $\lambda - \frac{n(T)}{T}$ after sufficiently large time $T$, where $n(T)$ is the number of outstanding demands at time $T$.

We focus our discussion on the single-vehicle setting, but results for multiple vehicle systems have been equally positive. Our simulations were of a gated, nearest-neighbor policy (NN); within each batch, the vehicle’s $k$th demand is the demand whose origin location was nearest to the vehicle at the time of delivery of the $(k-1)$th demand. Although a proof that a NN policy is perfectly stabilizing is currently not available, it has been observed that such policies have good performance for a variety of vehicle routing problems; they also have a fast implementation where large numbers of outstanding demands are concerned.

For each of Cases III and IV, Table 3.2 shows both the calculated threshold $\lambda^*$ (derived from the statistics in Table 3.1) and the estimate of $\lambda^*$ after time $T = 5000$. The simulated rate of arrivals $\lambda$ was unity in both cases. In both cases, the estimated and computed $\lambda^*$ were quite close (within 5% of each other).

<table>
<thead>
<tr>
<th>Case III</th>
<th>Case IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda^* = \frac{mv}{\mathbb{M}}$</td>
<td>$\approx 0.190$</td>
</tr>
<tr>
<td>$\lambda^*_T$, estimate after $T = 5000$</td>
<td>0.20</td>
</tr>
<tr>
<td>Error</td>
<td>$\approx 5%$</td>
</tr>
</tbody>
</table>

Table 3.2: Estimated stabilizable rate thresholds $\lambda^*$, for Cases III and IV, with a single unit-speed vehicle.

3.10 Conclusion

In this chapter, we have developed a general version of the 1-DPDP, as an on-line planning model for one-way Mobility-on-Demand systems with fleets of vehicles with limited carrying capacity. We analyzed routing policies for both the light traffic load and heavy traffic load cases, providing bounds on the average time spent by a customer in the system under these policies; the heavy-load analysis leveraged recent results, which we will discuss in Chapter 5, characterizing the lengths of optimal stacker crane tours through many random demands. We provided a new checkable
condition which is both necessary and sufficient to ensure that policies exist which can keep customer system times from growing unbounded (i.e., they stabilize). According to such condition, the Stacker Crane policy is stabilizing whenever stability is indeed possible. Moreover, we provided policy-independent lower bounds for the average system time, under certain conditions, which demonstrate that the Stacker Crane policy is efficient in certain cases (or almost efficient), in the sense that the rate of growth in the average system time matches the lower bound (or differs by a small factor). A number of open questions remain which require entirely new techniques to solve.

One significant limitation of our 1-DPDP model is that the vehicles can travel freely in $\mathbb{R}^d$, i.e., there are neither finite turning radii, nor “street constraints”. In Chapter 6, we will examine the DPDP instead when street constraints are imposed.

**Implementation issues:** The Stacker Crane policy requires on-line solutions to possibly large SCPs and TSPs, which are both NP-Hard problems. This suggests that there is no general algorithm capable of finding the optimal tour in a reasonable amount of time (e.g., polynomial in the number of demands). In practice, implementations of the Stacker Crane policy should rely on heuristics or on polynomial-time approximation schemes for the solutions to the SCPs and TSPs. Good heuristics for TSPs include Lin-Kernighan’s [76] or Christofides’ [34]. The LARGEARCS algorithm of [57] has been extremely effective for solving SCPs in practice; its runtime complexity is dominated by the cost of computing bipartite matchings as a subroutine. (In Chapter 5, we provide new formal proofs explaining the remarkable performance of LARGEARCS-like algorithms.) There are known algorithms for the Euclidean Bipartite Matching problem of order $O(n^{2+\epsilon})$ runtime, where $n$ is the size of the input and $\epsilon$ is an arbitrarily small positive constant [3].

Unfortunately, the performance of the Stacker Crane policy can be quite sensitive to the choice of matching heuristic. For example, in Chapter 5 we will find that constant-factor approximation, e.g., as guaranteed by the near-linear time ($O(n^{1+\epsilon})$) algorithm by Agarwal [2], may have dramatic effects on the performance achieved by the Stacker Crane policy, and may even compromise stability.

**Sources:** The findings of this chapter extend the author’s previous work on the 1-DPDP in [93] and [124]. [93] generalized the model presented in [119] to incorporate the more general demand generation model and multiple service vehicles; it allowed for a general spatial distribution of points, but required that $f_Y = f_X$. For that special case, it was shown correctly that $\bar{g} = \lambda \mathbb{E}_f \|Y - X\| / (mv)$. The SQM and Stacker Crane policies were presented with performance analysis. The lower bounds of Section 3.8 also appeared in [93], and the stability condition was first stated in terms of the simpler $g$. Of course, that stability analysis—and indeed the result itself—is no longer valid if $f_Y \neq f_X$. We gave the first extension of the 1-DPDP model to incorporate distributions with $f_Y \neq f_X$ in the last part of [124], and derived the new necessary and sufficient condition of Theorem 3.11. ([125] summarizes [93, 123, 124] in the form of a journal paper.)
Chapter 4

A Systematic Approach to Fleet-sizing in Practice: A Case Study in Singapore

4.1 Introduction

In this chapter we present part of a recent case study by Spieser et al. [114], about urban mobility, with financial justifications for shared-vehicle, Mobility-on-Demand systems. The study is particularly relevant because it employs insights from Chapter 3 to consider the fundamental issue of fleet sizing in a practical setting. The authors proposed a thought experiment whereby all modes of personal transport in a city are replaced by a fleet of shared autonomous vehicles, i.e., vehicles that are able to drive themselves in traffic, safely and reliably pick up passengers, and deliver them to their intended destinations. [114] mainly examines personal mobility in the country of Singapore, a small city-state for which transportation challenges are especially acute at the time of this writing.

Singapore is a fitting venue for consideration for several reasons: First, given the island’s diminutive size and high population density, officials are limited in the extent to which traditional measures (e.g., roadway expansion) can alleviate rising congestion. Second, despite Singapore’s sophisticated and well-subsidized public transportation system, the rate of private vehicle usage, and correspondingly traffic congestion, continues to increase. These factors make Singapore a promising candidate for conversion to shared-vehicle, autonomous Mobility-on-Demand (AMoD) systems. Finally, at the time the study was carried out, a rich collection of data pertaining to the country’s mobility demand was available, from which the statistics behind the analysis could be synthesized. Thus, using survey data about the population’s transportation demands, [114] suggests that autonomous MoD could meet the personal mobility need of the entire population of Singapore with a fleet whose size is approximately a third \( \frac{1}{3} \) of the total number of passenger vehicles in operation at the time of this writing. Moreover, combining such fleet-sizing guidelines with multi-faceted financial analysis, the study makes the case that AMoD systems are a financially viable alternative to
the current (i.e., private) means of personal mobility.

4.2 Literature Review: Shared-Mobility Systems

The efficiency gains that shared-vehicle systems can offer (both to the individual user and society as a whole) have been well documented [81]. For select cities, including Singapore, lists of these advantages have even been specifically compiled [47]. Eager to capitalize on emerging markets and to serve existing ones better, considerable effort has been devoted to characterizing the demand for shared-vehicle mobility therein [42]; such demand is affected by demographic [42, 28], and geographic considerations [33, 118], among others. Quality of service also plays an important (and delicate) role in mobility demand [67]: By fielding larger fleets, companies make it easier for patrons to rent a vehicle from a nearby station. This, in turn, draws new members to the program [79], requiring yet more vehicles to maintain the high level of service that initially attracted users.

One of the most prevalent contemporary solutions for Mobility-on-Demand is car-sharing, where customers check out cars from known locations for temporary usage. Financially, car sharing distributes the cost of purchasing, maintaining, and insuring vehicles across a large user-base, leveraging economies of scale to reduce the cost of personal mobility. It is well known that most private cars are used less than 10% of the time [81], so car sharing is a clear path towards sustainability. To date, the majority of car-sharing programs employ a round-trip vehicle rental model, where vehicles must be returned to the same station they were rented from. (Zipcar’s current rental service, for example, is based on this approach.) Noting the limitations in the round-trip rental model, one-way car-sharing services, such as car2go, have emerged. These services offer the added convenience of being able to return a vehicle to any one of multiple stations throughout the city [67]. However, left unchecked, asymmetries in travel patterns would, in general, create a surplus of vehicles at select stations, while leaving other stations under-served. Thus, rebalancing mechanisms are required to realign the supply of vehicles with the demand. Autonomous vehicles would provide perhaps the ideal such mechanism, to rebalance vehicle availability.

Simulation-based approaches have been used to infer the viability of various rebalancing schemes and, in turn, to gauge consumer demand for one-way car-sharing [86, 14, 15, 71, 48]. Initial findings suggest that one-way services are ideally suited for densely-populated urban centers. However, the lack of insight in the presence of a large number of relevant but uncertain parameters has been noted as a limitation of predominantly simulation-driven methods [67]. Thus, a more rigorous study has been pursued in several recent works, such as [91, 112] (and this thesis), aimed at understanding optimal rebalancing strategies, and the fundamental limits of performance in car sharing systems.

[114] applies insights from the latter study to the problem of fleet sizing in a practical, contemporary setting. [31] is similar to [114] in that it examines the problem of fleet-sizing quantitatively, for a hypothetical shared-vehicle system in three existing site in the United States (US). However, the approach therein is heavily simulation-
4.3 Fleet-Sizing Guidelines for AMoD Systems

4.3.1 Problem Formulation

The technical component of the study of fleet sizing in [114] is based on two main problems: (i) minimum fleet sizing, and (ii) fleet sizing with respect to minimum quality of service objectives. That is, the technical problems of the study are:

1. **Minimum fleet sizing**: What is the minimum number of vehicles necessary to serve all existing mobility demand?

2. **Performance-driven fleet sizing**: How many vehicles should be used to ensure that the quality of the service provided to customers (e.g., vehicle availability, or waiting time) is no less than a given standard?

The second problem acknowledges an intuitive trade-off between the fleet size and the user experience (beyond the bare minimum). We will present the methodology and results of the *minimum fleet sizing* analysis, because they are heavily motivated by the techniques developed in the thesis up to this point. The study of *performance-based fleet sizing* relies on alternative techniques, and so we refer interested readers to the source material and references therein.

Other benefits notwithstanding, financial considerations will undoubtedly factor into the decision of whether to switch to an AMoD system. [114] also presents a robust financial analysis, which, although essential to the formation of business strategy concerning AMoD systems, is beyond the scope of the thesis and is omitted.

4.3.2 Minimum fleet sizing

Model

Although the approach to the minimum fleet sizing problem in [114] is motivated by the insights of Chapter 3, it considers a version of the taxi theater with periodically time-varying demand and vehicle models. While substantially more difficult to analyze rigorously, time-varying models are far more appropriate to capture, e.g., (i) the characteristic of urban mobility demand throughout a whole day (24 hours), and (ii) vehicle speeds under various levels of congestion during the same time.

**Euclidean taxi theater with time-varying components**: Transportation demands form a random process \(\{(t_i, X_i, Y_i)\}_{i \in \mathbb{N}}\) such that: the \(i\)-th transportation demand is revealed at time \(t_i\) and poses the requirement to travel from origin point \(X_i\) to destination point \(Y_i\). Trip requests are to be serviced by vehicles that may transport at most one demand at a time, moving within a compact planar region (environment) \(\mathcal{W} \subset \mathbb{R}^2\) (e.g., on the road-system of Singapore); thus, note that \(X_i, Y_i \in \mathcal{W}\). The vehicles’ top speed \(v(t)\) is given but periodically time-varying, e.g., due to congestion.
(While indeed such limits due to congestion may themselves be affected by routing choices, for simplicity, the study does not consider this dependence explicitly.) The arrival times \( \{t_i\} \) form a non-stationary temporal Poisson process with rate function \( \lambda(t) \). The trip data \( \{(X_i, Y_i)\} \) are independent of each other; however, the \( i \)-th O-D pair \( (X_i, Y_i) \) is conditionally distributed according to \( f(\cdot; t_i) \), for a given probability distribution function \( f \), called the demand distribution. Thus, e.g., the expected number of demands revealed within a time interval \([t_1, t_2]\), and with \( X_i \in \mathcal{W}_1 \) and \( Y_i \in \mathcal{W}_2 \), for any times \( t_1 \) and \( t_2 \) and regions \( \mathcal{W}_1, \mathcal{W}_2 \subseteq \mathcal{W} \), is

\[
\int_{t_1}^{t_2} \lambda(t) \int_{(p,q) \in \mathcal{W}_1 \times \mathcal{W}_2} f(p, q; t) \, dp \, dq \, dt.
\]

The authors partitioned the day into 24 hour-long intervals or “bins”, and supposed that the rate function \( \lambda(t) \) and distribution function \( f(\cdot; t) \) are constant \( (\lambda_k \text{ and } f_k, \text{respectively}) \) within any bin \( k \).

**Analysis**

Using the intuition behind Theorem 3.11 of Chapter 3, [114] reasoned that a fleet of \( m \) vehicles can keep up with user demand only if the fleet, as a whole, is able to cover distance at least as quickly on average as the rate at which such “work” accumulates. Supposing that nearly all trips are completed during the time of the bin in which they arrive, one can argue that approximately

\[
\sum_k (\lambda_k T_k) \mathcal{M}_k
\]

average total service distance enters the system per day, where \( T_k \) is the duration (e.g., one hour) of bin \( k \), and \( \mathcal{M}_k := \mathcal{M}(f_k) \) is the Mover’s complexity of the associated O/D distribution. That is, approximately \( \lambda_k T_k \) demands, times approximately \( \mathcal{M}_k \) distance per demand, is required for each bin \( k \). A fleet of \( m \) vehicles, each capable of travelling at average speed \( v_k \) during the \( k \)-th bin, is able to cover distance at a rate of \( m \sum_k v_k T_k \) daily. Therefore, for the service rate to exceed the rate of work, it requires (at a minimum) that

\[
m > \sum_k (\lambda_k T_k) \mathcal{M}_k / \sum_k v_k T_k. \tag{4.1}
\]

**4.4 Data Sources**

In order to obtain the statistical models which are the subject of analysis of Section 4.3.2, three complementary data sources (described below) were leveraged.

**The Household Interview Travel Survey** — The Household Interview Travel Survey, or simply HITS, is a comprehensive survey conducted periodically by the Land Transport Authority (LTA) for the purpose of gathering an overview of high-level transportation patterns within Singapore [107]. This work employed the 2008
HITS survey in which 10,840 of the then 1,144,400 households in Singapore were selected to participate in the survey. The HITS database, which summarizes the survey, is structured as follows. For each household surveyed, each resident reported specific details of each trip taken on a recent weekday of interest. In general, each trip is comprised of several stages with a new stage introduced each time the participant switched their mode of transport, e.g., transferred from the subway to bus as part of the same trip. For each trip, the resident reported the trip’s origin point, destination point, start time, end time, and the mode of transport, e.g., car, bus, subway, etc., used in each sub-stage. From the HITS data, 56,839 distinct trips were extracted. After eliminating trips for which GPS coordinates had been compromised, 56,673 trips remained.

**Singapore Taxi Data**—To gather ground truth traffic characteristics, we rely on a database of taxi records collected over the course of a week in Singapore in 2012. The data chronicles the movement and activities of approximately 60% of all active taxis by recording each vehicle’s GPS coordinates, speed, and passenger status, e.g., “passenger-on-board,” “vacant,” “responding to call,” etc. Owing to the high rate at which recordings are taken, approximately every 30 seconds to 1 minute per vehicle, and the large number of taxis contributing to the database (more than 10,000), the fleet, collectively, serves as a distributed, mobile, embedded traffic sensor which may be queried to provide an estimate of traffic conditions throughout the city.

**Singapore Road Network**—A graph-based representation of Singapore’s road network is used to determine the most efficient routes automated vehicles should take from point to point in Singapore (whether carrying a passenger or moving to fetch one). When the analysis method required simpler distance evaluations, the average ratio of trip length versus straight-line (Euclidean) distance was estimated from the taxi data as a factor $\beta = 1.38$.

## 4.5 AMoD Fleet Sizing for Singapore

To apply the analysis of Section 4.3.2 to the data of Section 4.4 requires a number of relevant statistics to be estimated. In this section we describe the methods used to obtain such statistics from the three disparate data sources. Also, we provide the numerical values (in plots) obtained.

**Arrival Rate** $(\lambda)$—Let $\lambda^\text{HITS}_k$ denote the average arrival rate of trips in hour $k \in \{0, 1, \ldots, 23\}$, based solely on the HITS survey. The overall arrival rate in hour $k$ is estimated as $\lambda_k = \alpha \lambda^\text{HITS}_k$, where $\alpha = 1, 144, 400/10, 840 \approx 105.57$ is the scaling factor that, inversely, reflects the fraction of the households that took part in the HITS survey. The rates $\{\lambda_k\}$ estimated in this way are plotted as the light-blue line in Figure 4-1.

**Mean Carry Distance** The mean carry distance (O-D distance) is the first term of the Mover’s complexity in (3.3). The average carry distance within each hour $k$ was used as an estimate of the mean carry distance for the hour. The calculation assumed each trip took place on the shortest path (as measured by distance) between origin and destination. Shortest path algorithms, e.g., Dijkstra’s algorithm,
are computationally efficient, allowing calculations to be run on a detailed roadmap of Singapore. The average such O/D distance ranged, on an hourly basis, from a minimum of 6.47km to a maximum of 13.31km. It is plotted as the dark-blue line in Figure 4-1.

**Earth Mover’s Distance**—To estimate the EMD (second term), Singapore was partitioned into a number of smaller regions, thus defining pick-up and drop-off bins. Origin and destination points of trips were assigned to bins accordingly, and the EMD was approximated using a min-cost flow formulation (Problem 5.1) discussed in detail in Chapter 5. A scaling factor \( \beta = 1.38 \) was used to approximate distances on the underlying roadmap, given straight-line distances between regions. The hourly estimates of Mover’s complexity \( \{M_k\} \) are plotted as the green line in Figure 4-1. (An alternative method, which could potentially estimate the EMD much more accurately, would be to use a new formulation of the EMD discussed at length in Chapter 8, on the following hourly estimate of the demand distribution function.)

**Mobility Demand Distribution** \((f)\)—The road network of Singapore was divided into road segments, each of length no greater than 6km. Each pair of such segments was treated as a \((doubly) \) spatial bin, and a trip from hour \( k \) was assigned to bin \((a, b)\) if its origin was on segment \( a \) and its destination was on segment \( b \). For each hour \( k \), the estimated demand distribution \( f_k \) was taken as the distribution whose generative model is: (i) choose a bin \((a, b)\) with probability proportional to the number of hour \( k \) trips assigned to it, then (ii) sample O/D pair \((X, Y)\), with \( X \) and \( Y \) independent and uniformly distributed, over segments \( a \) and \( b \), respectively.

**Average Velocity** \((v)\)—Taxi data was used to determine a conservative estimate of the average speed at which occupied taxis move about the city, under current traffic conditions. This value was then used as an estimate of \( \{v_k\} \) in (4.1). (Note that such estimate does not take into account potential changes in congestion due to vehicle sharing.) To determine how fast the average laden taxi travels, one divides the total distance travelled by a taxi with a passenger on board by the total time spent by the taxi with a passenger on board (that is, on an hourly basis) over the course of a typical week. \( \{v_k\} \) is plotted as the red line in Figure 4-1.

Given all aforementioned quantities, (4.1) yields that at least 92,693 automated vehicles are required to ensure that the vehicle fleet can satisfy the existing demand. Note however, that this figure should be seen as a lower bound on the fleet size, since customer waiting times would be unacceptably high.

### 4.6 Summary of Performance-driven Fleet Sizing

[114] employed simulation-based techniques, described in its Sections 3.3 and 5.2, to consider how much the fleet size should be increased in order to reduce the waiting times of customers to acceptable levels. The demand model for the simulation was derived from the data of Section 4.4 using a procedure described in its Section 5.2.

Figure 4-2 shows simulation results of average wait times over the course of a day. For 250,000 vehicles, the maximum wait times during peak hours is around 30 minutes, which is comparable with typical congestion delays during rush hour. With
Figure 4-1: Summary of data-derived statistics, used to estimate the minimum fleet size. According to (4.1), the minimum fleet size to serve all of Singapore’s mobility demand is 92,693 shared vehicles. At 1,144,400 households in Singapore, that would be roughly one shared car for every 12.3 households.

300,000 vehicles, peak wait times are reduced to less than 15 minutes. To put these numbers into perspective, in 2011 there were 779,890 passenger vehicles operating in Singapore [74].

4.7 Conclusion and Additional Insights

Based on intuition developed from the insights in Chapter 3, [114] provided analytical guidelines for sizing Automated Mobility-on-Demand (AMoD) systems using actual survey data. Other simulation-based results suggest that an AMoD solution could meet the personal mobility need of the entire population of Singapore with a fleet whose size is approximately 1/3 of the total number of passenger vehicles currently in operation. Such estimates correspond to a vehicle sharing ratio of approximately 3.5–4.5. Moreover, financial analysis indicates AMoD systems are a financially viable alternative to more traditional means of accessing personal mobility.

[114] also presents insights about the relative value of AMoD in different settings. For example, on a relative basis, the savings afforded by AMoD technology in Singapore stem largely from the ability to split the hefty cost of car ownership. In the US, the savings are predominantly the result of being able to travel more comfortably and eliminate parking activities.

Finally, [114] suggested a number of issues about AMoD that deserve further investigation. An important aspect that needs to be addressed is the impact of an
AMoD system on traffic congestion. Even though the analysis shows that an AMoD system could provide mobility to the entire population with far fewer vehicles than are currently on the road, it is also the case that such vehicles would be travelling more; in fact, the total distance travelled by all vehicles (often referred to as Vehicle Miles Traveled, or VMT)—and thus the load on the road network—will be greater, due to vehicles travelling empty, i.e., to fetch customers. Another important aspect is latent demand: it may be the case that the availability of a new convenient and economical mode of transportation may actually increase the demand for mobility. Given these competing forces, it is as yet unclear what the effect of AMoD systems will be on travel times and congestion levels, which is an important topic for future research.

Acknowledgements: The author would like to thank co-authors Kevin Spieser and Emilio Frazzoli at MIT, Rick Zhang and Marco Pavone at Stanford, and Daniel Morton, with the Singapore-MIT Alliance for Research and Technology, for the opportunity to apply the theory developed in this thesis in a practical setting. All authors remain grateful to the Land Transport Authority of Singapore for providing access to HITS data.

Figure 4-2: Average wait times in simulation, as a function of time (periodic), for hypothetical fleet sizes.
Chapter 5

Cost Bounds and Asymptotically Optimal Algorithms for the Euclidean Stacker Crane Problem with Random Demands

5.1 Introduction

Chapter 3 demonstrated the key role of the Stacker Crane problem both in solving and in analysing the 1-DPDP. Recall that the Stacker Crane problem is the static problem of finding the shortest tour by which a single vehicle of unit carrying capacity can satisfy a collection of demands described by origin/destination (O/D) pairs. The SCP is an NP-Hard problem, meaning it is unlikely that algorithms exist which can compute optimal tours in less than exponential time in terms of the number of demands. (The SCP is NP-Hard even if the instances are generated by O/D-pairs in Euclidean space.)

5.1.1 Contributions

In this chapter, we embed the Stacker Crane problem within a probability framework, where origin/destination pairs are identically and independently distributed within an Euclidean environment. Our random model is general in the sense that we consider potentially non-uniform distributions of points, with an emphasis on the case that the marginal distributions of origins and destinations are distinct. We derive the results about the resulting stochastic SCP which were leveraged heavily for analysis of the 1-DPDP in Chapter 3.

Under the given probability framework, we show that the LARGEARCS heuristic (one of two components of the well-known Frederickson-Hecht-Kim (FHK) approximation algorithm [57]) is asymptotically optimal almost surely. That is, the ratio L.ARCS(n)/SCP*(n) goes to 1 as n → +∞, except on a set of outcomes of zero probability; here, L.ARCS(n) denotes the length of the stacker crane tour generated...
by LARGEARCS (on $n$ random demands), and $\text{SCP}^*(n)$ denotes the length of the optimal stacker crane tour. While the phenomenally good performance of LARGEARCS for large instances has been widely acknowledged in practice, we are not aware of any previous, formal guarantees of optimality. The best guarantees previously known are that FHK computes tours which are $9/5$-optimal in the worst case.

We embed LARGEARCS in a class of polynomial-time approximation algorithms, which we will call SPLICE. We demonstrate that all of the SPLICE algorithms obtain optimality, in the same sense, if the origins and destinations are independent, i.e., $Y \perp X$. From a technical standpoint, our result leverages a novel connection between the Euclidean Bipartite Matching problem and the theory of random permutations. (Similar results may also hold in the case $Y \not\perp X$, but would require extra study.)

In addition to the proofs of optimality, we provide strong asymptotic cost bounds, which hold for (i) the LARGEARCS tour, (ii) the optimal stacker crane tour, and (iii) the SPLICE algorithms in the case $Y \perp X$. From a technical standpoint, the bounds leverage a connection between the SCP and the Euclidean Bipartite Matching problem. The main technical challenge has been to generalize known bounds for bipartite matchings, i.e., Theorems 2.14 and 2.15, to scenarios where the two types of points have distinct distributions. (We believe that these bound hold for all the algorithms in SPLICE, regardless of the $Y \perp X$ assumption; however, we have not been able to prove it.)

Discussion: The immediate importance of our result is that it assures us that the Stacker Crane policy of Chapter 3 may be implemented using efficient heuristics, without significant consequences in terms of performance, e.g., to stability and system time bounds. While the FHK algorithm has always enjoyed a guarantee of $9/5$-optimality in the worst case, constant-factor guarantees introduce a gap in Theorem 3.11 between necessary and sufficient conditions for stability. Such gap occurs because stacker crane tours grow at the rate $\Theta(n)$ in the number of demands, and not sub-linearly, e.g., like TSP tours. Thus, in contrast, the effects of substituting approximately optimal TSP tours (which grow at a rate $\Theta(n^{(d-1)/d}) = o(n)$) in DTRP policies have been, typically, fairly benign. For example, using constant-factor tours in TSP-based routing policies (e.g., those in [22]), result in system time bounds worse usually by at most a constant factor.

5.1.2 Organization

This chapter is structured as follows. In Section 5.2 we provide some background on heuristics to solve the Stacker Crane problem. In Section 5.3 we rigorously state the stochastic SCP and the chapter objectives, i.e., to obtain a computationally efficient, asymptotically optimal algorithm and strong cost bounds. In Section 5.4 we analyze LARGEARCS and its embedding in SPLICE, a class of algorithms for the stochastic SCP that run in polynomial-time and tend to perform well. In Section 5.5 we derive a set of analytical bounds on the lengths of tours produced by LARGEARCS, SPLICE algorithms, and the optimal solution. In Section 5.6 we present simulation experiments to empirically validate our claims of optimality and cost bounds. Finally, we present concluding remarks in Section 5.7.
5.2 Solving the Stacker Crane Problem

Literature overview. The SCP, being a generalization of the Traveling Salesman Problem (TSP), is NP-Hard [58]. The problem is NP-Hard even on trees, since the Steiner Tree Problem can be reduced to it [56]. (The problem, however, is in P on paths [11].) In [56], the authors present several approximation algorithms for tree graphs with worst-case performance ratio ranging from 1.5 to around 1.21. The 1.5 worst-case algorithm, based on a Steiner tree approximation, runs in linear time. Recently, one of the polynomial-time algorithms presented in [56] has been shown to provide an optimal solution on almost all inputs (with a 4/3-approximation in the worst case) [36]. An average case analysis of the SCP on trees has been examined for the special case of caterpillars as underlying graphs [37].

Approximation. For general graphs, the best approximation ratio is $9/5$ and is achieved by an algorithm by Frederickson et al. [57]; we will call it FHK. FHK actually consists of running two algorithms, LARGEARCS and SMALLARCS, and the final solution is the smaller of the two results. Lapsing momentarily to the notation of [57], if we let $C^*$ denote the cost (length) of the optimal SCP tour, and $C_A$ denote the total length of only the origin-to-destination parts (a tour-independent constant), then the LARGEARCS algorithm itself is $(3 - 2C_A/C^*)$-optimal, and the SMALLARCS algorithm is $(3/2 + (1/2)C_A/C^*)$-optimal; the $9/5$ worst-case factor of FHK is the tight uniform upper bound over the minimum of the two factors. LARGEARCS has runtime complexity $O(n^3)$ in the number of O/D pairs; in the planar case, it has complexity $O(n^{2+\varepsilon})$, for arbitrarily small positive constant $\varepsilon$.

5.3 Problem Statement

In this chapter we study the SCP on subsets of the random i.i.d. sequence of demands $\{(X_k, Y_k)\}$ with general distribution $f$ of O/D pairs in a compact, Euclidean workspace $W$.

We aim at solving two problems in the chapter:

Prob. 1 Find at least one polynomial-time algorithm $A$ for the ESCP which is asymptotically optimal almost surely, i.e.,

$$\lim_{n \to +\infty} \frac{A_f(n)}{SCP_f^*(n)} = 1^+,$$

where $n$ is the size (number of demands) of the stochastic instance, $A_f(n)$ is the length of the stacker crane tour produced by algorithm $A$, and $SCP_f^*(n)$ is the length of the optimal stacker crane tour.

Prob. 2 For the general case (i.e., $f_Y = f_X$ not necessary), characterize the rate of growth of the optimal tour length $SCP_f^*(n)$.

The solutions to the above problems lead to a robust class of polynomial-time, provably-efficient algorithms for the SCP, and useful bounds on the rate of growth of the optimal SCP tour.
5.4 Asymptotically Optimal Polynomial-Time Algorithms for the Stochastic ESCP

5.4.1 LARGEARCS and the SPLICE Algorithms

A key property of stacker crane tours is that they visit the origins and destinations of the demand set in a strictly alternating pattern; i.e., a vehicle moves either occupied from an origin to the corresponding destination, or else unoccupied from a destination to some (remaining) origin. The tour fragments from origin to destination are essentially determined by the problem data through shortest paths. The remaining fragments consist of links from each destination to a subsequent origin; clearly, each link goes to a different origin. The SCP is to choose the links which produce the shortest, single, closed tour.

The main idea of the LARGEARCS heuristic is to route the vehicle from destinations back to origins using the set of destination-to-origin links with the smallest total length. The task is accomplished by bipartite matching. However, when the origin-to-destination links are added back in, the result is quite unlikely to form a single closed tour; instead, there will be a certain number of disconnected subtours (see Figure 5-1(b)). Thus, in general, the result is not a stacker crane tour.

Nevertheless, clever heuristics can be used to connect such subtours to form a stacker crane tour. For example, the LARGEARCS algorithm connects subtours opportunistically while following a 2-optimal TSP tour through the set of origins (generated using the popular minimum spanning-tree (MST) heuristic, i.e., doubling up the edges in the MST); by "opportunistically" we mean that it only follows segments of the TSP tour when it is not incident to an uncompleted subtour. Other subtour-connecting heuristics are possible, and it is in this spirit that we define the class of algorithms SPLICE. The SPLICE template is given in pseudo-code below.

Algorithm SPLICE

Input: a set of demands $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}, n > 1$, and a "rewiring" algorithm REWIRE.
Output: a stacker crane tour $P$ through the demand set $\mathcal{D}$.

1: $\sigma \leftarrow$ the optimal Euclidean bipartite matching between sets $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ and $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$, computed using an algorithm $\mathcal{M}$.
2: $Q \leftarrow \emptyset$.
3: Add the $n$ origin-to-destination tour fragments $x_i \rightarrow y_i, i = 1, \ldots, n$ to $Q$.
4: Add the $n$ matching links $y_i \rightarrow x_{\sigma(i)}, i = 1, \ldots, n$ to $Q$.
5: Run REWIRE with inputs $\mathcal{D}$ and $Q$, and return the resulting tour $P$.

In line 1, the algorithm $\mathcal{M}$ may be any algorithm that computes optimal bipartite matchings. After the origin-to-destination links are added (line 3) and the optimal bipartite matching links are added (line 4), there might be a number $N$ of disconnected subtours. In that case (i.e., when $N > 1$) the subtours must be cut open and additional links must be added, to connect them into a single stacker crane tour.
(a) Line 3: Six origin-to-destination links are added.

(b) Line 4: Six matching links are added. The number of disconnected subtours is \( N = 2 \).

(c) Links \( y_3 \rightarrow x_1 \) and \( y_5 \rightarrow x_6 \) are replaced by links \( y_3 \rightarrow x_6 \) and \( y_5 \rightarrow x_1 \), respectively (by some algorithm REWIRE), and the tour is completed.

Figure 5-1: Sample execution of a SPLICE algorithm. The solution to the EBMP is \( \sigma(1) = 2, \sigma(2) = 3, \sigma(3) = 1, \sigma(4) = 5, \sigma(5) = 6, \) and \( \sigma(6) = 4 \). Demands are labeled with integers. Origin and destination sites are represented by solid and dashed circles, respectively. Origin-to-destination links are shown as black arrows. Matching links are dark dashed arrows. Subtour connections are shown as lighter, dashed arrows. The resulting tour is \( 1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 4 \rightarrow 5 \rightarrow 1 \).
SPLICE is parameterized by the choice of the rewiring algorithm; one may choose any “rewiring” algorithm satisfying the following basic properties:

1. First, given the demands \( D \), and a set of subtours \( Q \) generated by a permutation \( \sigma \) in lines 3-4, the algorithm must produce a stacker crane tour.

2. Furthermore, at most \( N \) links may be inserted into \( Q \), where \( N \) is the number of subtours originally in \( Q \). (Note: that under the previous two constraints, at most \( N \) link deletions will be possible as well.)

3. Finally, the algorithm should have polynomial runtime complexity; ideally, no worse than that of the Euclidean bipartite matching algorithm \( \mathcal{M} \); e.g., in the planar case, to beat the Agarwal algorithms \([3]\), it should be \( O(n^{2+\varepsilon}) \) for some (all?) \( \varepsilon > 0 \).

Regarding notation, suppose that an algorithm called REWIRE is chosen: Then we will denote by SPLICE+REWIRE the resulting algorithm for SCP. LARGEARCS is the member for which REWIRE is the 2-optimal minimum spanning-tree TSP heuristic, with shortcutting. (\( \mathcal{M} \) is technically also a parameter of the class. However, since \( \mathcal{M} \) must choose the optimal matching, it only affects runtime, and not performance.

To simplify the discussion, let us assume \( \mathcal{M} \) is an arbitrary polynomial-time bipartite matching algorithm, but fixed.) Where the results are not influenced by the particular choice of REWIRE, we will simply use the name SPLICE, with the understanding that the statement will hold for any member of the class. For example, by stating that SPLICE (the class) is asymptotically optimal, we would mean that every algorithm in SPLICE is asymptotically optimal. Figure 5-1 illustrates a typical execution of a SPLICE algorithm; we refer to the destination-to-origin links added in Figure 5-1(c) as connecting links, since they connect the subtours.

### 5.4.2 Asymptotic Optimality of LARGEARCS

**Theorem 5.1 (Asymptotic optimality of LARGEARCS).** For any compact, dimension \( d \geq 1 \) Euclidean workspace \( \mathcal{W} \), and any sequence of independent random O/D pairs \( \{(X_k, Y_k)\} \), with distribution \( f \) over \( \mathcal{W}^2 \), the length of the stacker crane tour through the first \( n \) demands, \( \mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\} \), given by the LARGEARCS heuristic, has the limit behavior

\[
\lim_{n \to +\infty} \frac{\text{L.ARCS}(n; f)}{\text{SCP}^*(n; f)} = 1, \quad \text{almost surely.}
\]

**Proof.** A stacker crane tour is composed of origin-to-destination links and destination-to-origin links. The latter describe some bipartite matching between origins and destinations having total cost no less than that of the optimal bipartite matching. Thus, one can write

\[
\text{SCP}^*(n; f) \geq \sum_{i=1}^{n} \|Y_i - X_i\| + \mathcal{M}^*(Y_n, X_n),
\]

82
where \( \mathcal{X}_n \) is the set of \( n \) origins, and \( \mathcal{Y}_n \) is the set of \( n \) destinations. By the Strong Law of Large Numbers, \( n^{-1} \sum_{i=1}^{n} \| Y_i - X_i \| \to \mathbb{E}_f \| Y - X \| \), almost surely; thus, \( \text{SCP}^*(n) \) is bounded below by linear growth. The right-hand side of (5.1) is the total length of the subtours generated by lines 3-4 of SPLICE. The LARGEARCS tour is larger by at most twice the length of the optimal TSP tour through \( \mathcal{X}_n \), i.e.,

\[
\text{L.ARCS}(n; f) \leq \sum_{i=1}^{n} \| Y_i - X_i \| + M^*(\mathcal{Y}_n, \mathcal{X}_n) + 2 \text{TSP}^*(\mathcal{X}_n).
\]

Thus, we have

\[
\frac{\text{L.ARCS}(n; f)}{\text{SCP}^*(n; f)} \leq 1 + 2 \frac{\text{TSP}^*(\mathcal{X}_n)}{\text{SCP}^*(n; f)} \leq 1 + 2 \frac{\text{TSP}^*(\mathcal{X}_n)/n^{(d-1)/d}}{n^{-1} \text{SCP}^*(n; f)} \times \frac{n^{1-1/d}}{n}. \]

Taking the \( \limsup_{n \to +\infty} \) on both sides and observing (i) \( \lim_{n \to +\infty} \text{TSP}^*(\mathcal{X}_n)/n^{(d-1)/d} = 0 \), (ii) \( \liminf_{n \to +\infty} n^{-1} \text{SCP}^*(n; f) > 0 \), and (iii) \( n^{-1/d} \to 0^+ \), we obtain the result. (The \( \liminf \) is obviously at least 1.)

**Discussion**

It is clear, by extension of Theorem 5.1, that FHK is asymptotically optimal, since the FHK tour is always at least as good as the LARGEARCS one. As far as the author knows, no such probabilistic guarantees have been proved previously.

**The case against SMALLARCS:**

In contrast to LARGEARCS, the SMALLARCS component of FHK seems to be increasingly superfluous as the number of demands \( n \) increases. Casual inspection of the algorithm suggests that for large \( n \) it is actually quite unlikely to produce a solution of factor close to one. For a brief justification of this statement, we offer the following observations: Roughly speaking, the SMALLARCS algorithm works in two phases. First, it finds a SCP-like tour through the demands, except it allows O/D pairs to be served in either the forward or reverse order (i.e., by visiting the destination site before the origin site). In the second phase, all demands serviced in reverse order are “corrected” with additional links between the origin and destination sites. For SMALLARCS to achieve a factor near one, nearly all demands would have to be served in the proper order by the initial tour. Such an event should be increasingly statistically unlikely as \( n \) increases. We will put our claim to an empirical test in Section 5.6.

Another argument against the SMALLARCS algorithm is a difference in runtime. The runtime of LARGEARCS (or SPLICE in general) is dominated by the bipartite matching of origin and destination sites. Euclidean bipartite matching can be computed in \( O(n^{2+\epsilon}) \) time. SMALLARCS is dominated by an all-pairs shortest paths algorithm (\( O(n^3) \) time, e.g., by Floyd-Warshall [40]), and by Christophides’ TSP al-
algorithm [34] \(O(n^3)\) time). In other words, SMALLARCS involves a factor nearly \(n\) extra runtime.

### 5.4.3 Asymptotic Optimality of SPLICE under \(Y \perp X\)

In this section we focus on the case that destination points of a demand are independent of the origins, i.e., on the case \(Y \perp X\). In this albeit unlikely scenario, we show that all of the SPLICE algorithms obtain the same asymptotic optimality guarantee as LARGEARCS. In general, the re-wiring algorithm given to SPLICE will have to add a number of connecting links between disconnected subtours (since, in general \(N > 1\)). However, when \(Y \perp X\), we can prove that the number of disconnected subtours grows quite slowly \(O(\log n)\) in the number of demands. Thus, as long as the connecting procedure satisfies the rather benign restrictions of SPLICE, the total cost of the extra links cannot be too large.

**Lemma 5.2 (Number of subtours).** For any compact, dimension \(d \geq 1\) Euclidean workspace \(W\), and any sequence of independent random O/D pairs \{(X_k, Y_k)\}, with distribution \(f\) over \(W^2\) and \(Y_k \perp X_k\), the number \(N_n\) of subtours generated by lines 3-4 of SPLICE, on the first \(n\) pairs, has limit behavior \(\lim_{n \to +\infty} N_n/n = 0\), almost surely.

The proof of Lemma 5.2 requires some intermediate results, which we will introduce shortly. However, given the Lemma, it is quite trivial to show that SPLICE algorithms are asymptotically optimal.

**Theorem 5.3 (Asymptotic optimality of SPLICE).** For any compact, dimension \(d \geq 1\) Euclidean workspace \(W\), and any sequence of independent random O/D pairs \{(X_k, Y_k)\}, with distribution \(f\) over \(W^2\) and \(Y_k \perp X_k\), the stacker crane tour given by any SPLICE algorithm has limit behavior

\[
\lim_{n \to +\infty} \frac{\text{SPLICE}(n; f)}{\text{SCP}^*(n; f)} = 1^+, \quad \text{almost surely.}
\]

**Proof.** The proof is essentially the same as the proof of Theorem 5.1. Since SPLICE adds a number at most \(N_n\) of connecting links to the subtours generated by lines 3-4, the SPLICE tour is at most \(N_n\) \(\text{diam} W\) longer than the total length of the subtours. One obtains the result by mimicking the logic of the proof of Theorem 5.1, with \(N_n/n = o(1)\) almost surely.

In order to obtain the order of growth of the number of generated subtours, first we observe an equivalence between the number of subtours \(N_n\) produced by lines 3-4 of SPLICE and the number of cycles of the permutation \(\sigma\) in line 1.

**Lemma 5.4 (Permutation cycles and subtours).** The number \(N_n\) of subtours produced by lines 3-4 of SPLICE is equal to \(N(\sigma)\), the number of cycles of the permutation \(\sigma\) in line 1.
Proof. Let $Y_k$ be the set of destinations in subtour $k$ ($k = 1, \ldots, N$). By construction, the indices in $Y_k$ constitute a cycle of the permutation $\sigma$. For example, in Figure 5-1, the indices of the destination sites in the subtour $x_1 \rightarrow y_1 \rightarrow x_2 \rightarrow y_2 \rightarrow x_3 \rightarrow y_3 \rightarrow x_1$ are $\{1, 2, 3\}$, and they constitute a cycle for $\sigma$ since $\sigma(1) = 2$, $\sigma(2) = 3$, and $\sigma(3) = 1$. Since the subtours are disconnected, and every index is contained by some subtour, then the sets $Y_k$ ($k = 1, \ldots, N$) represent a partition of $\{1, \ldots, n\}$ into the disjoint cycles of $\sigma$. This implies that the number of subtours $N_s$ is equal to $N(\sigma)$.

By the Lemma above, the number of subtours generated by lines 3-4 of SPLICE is equal to the number of cycles of the permutation $\sigma$. Leveraging the i.i.d. structure in our problem setup, one can argue intuitively that all permutations should be equiprobable. In fact, the statement withstands rigorous proof.

Lemma 5.5 (Permutations equi-probable). For any compact, dimension $d \geq 1$ Euclidean workspace $W$, and any sequence $\{(X_k, Y_k)\}$ of independent random O/D pairs, with distribution $f$ over $W^2$ and $Y_k \subseteq X_k$, the permutation in line 1 of SPLICE on the first $n$ pairs has distribution 

\[ P[\sigma] = 1/n! \quad \text{for all } \sigma \in \Pi_n. \]

Proof. See Appendix B.1 for the complete proof.

Lemmas 5.4 and 5.5 allow us to apply Prop. 2.9 to characterize the growth order for the number of subtours; in particular, we observe that $\mathbb{E}N_n = \mathbb{E}N(\sigma) = \log(n) + O(1)$. We are now equipped to prove Lemma 5.2.

Proof of Lemma 5.2. For any $\epsilon > 0$, consider the sequence $E$ of events, where

\[ E_n = \left\{ (X_n, Y_n) : N_n/n > \epsilon \right\} \]

or, equivalently, $E_n = \left\{ (X_n, Y_n) : (N_n - \mathbb{E}N_n) + (\mathbb{E}N_n - \log(n)) + \log(n) > \epsilon n \right\}$. By Lemma 5.4, the number of disconnected subtours is equal to the number of cycles in the permutation $\sigma$ computed by the matching algorithm $M$ in line 1. Since, by Lemma 5.5, all permutations are equiprobable, the number of cycles has expectation and variance both equal to $\log(n) + O(1)$. Therefore, we conclude that $N_n$ has expectation and variance both $\log(n) + O(1)$. Hence, we can rewrite the events $E_n$ as:

\[ E_n = \left\{ (X_n, Y_n) : N_n - \mathbb{E}N_n > \epsilon n + o(n) \right\}. \]

Applying Chebyshev’s inequality, we obtain (for any $n'$ sufficiently large, yet finite)

\[ \sum_{n=0}^{\infty} P[E_n] \leq n' + \sum_{n=n'}^{\infty} \frac{\log(n) + O(1)}{[\epsilon n + o(n)]^2}. \]

Since this summation is finite, we can apply the Borel-Cantelli lemma to the sequence of events $E_n$ and conclude that $\mathbb{P}[\limsup_{n \to +\infty} E_n] = 0$. Finally, since $\epsilon$ can be chosen arbitrarily small, the upper limit of the claim follows (the lower limit holds trivially).

85
Remark 5.6. For \( d \geq 3 \), one can similarly prove that \( \lim_{n \to +\infty} N_n / n^{(d-1)/d} = 0 \) almost surely.

Discussion: The consequence of log-number of subtours is it shows that the SCP functional has asymptotic behavior that is tightly related to that of the bipartite matching functional. In the sequel, this connection will allow us to leverage a deep but incomplete set of probabilistic results for bipartite matching [5, 120, 121, 46] to obtain new cost bounds for bipartite matchings and stacker crane tours.

5.5 Analytical Bounds on the Cost of the ESCP

In this section we derive analytical bounds on the length of the optimal stacker crane tour through a random set of demands obeying the probability framework described in Section 3.4. The resulting bounds are useful for two reasons: (i) they give further insight into the ESCP (and the EBMP), and (ii) they allowed us to derive the stability condition and performance bounds for the simple model (1-DPDP) and routing policies of the previous chapter (Chapter 3).

First, in Section 5.5.1, we derive a lower bound on the total cost of the Euclidean matching for the case \( f_Y \neq f_X \) (also, a resulting lower bound for the ESCP); both scale linearly in the number of O/D pairs. Then, in Section 5.5.2, we find upper bounds whose linear parts match the rates of the lower bounds.

5.5.1 Lower Bounds on Lengths of Euclidean SCP Tours

Suppose we have a finite partition \( \mathcal{C} = \{C^1, \ldots, C^{|\mathcal{C}|}\} \) of the Euclidean workspace \( \mathcal{W} \) into \( |\mathcal{C}| \) cells. We will denote by

\[
f_X(C^i) := \mathbb{P}[X \in C^i] = \int_{C^i} d f_X(x)
\]

the probability of cell \( C^i \) under the origin distribution \( f_X \), i.e., the probability that a particular origin \( X \) is in the \( i \)-th cell. Similarly, we denote by

\[
f_Y(C^i) := \mathbb{P}[Y \in C^i] = \int_{C^i} d f_Y(x)
\]

the cell’s probability under the destination distribution \( f_Y \), i.e., the probability that a particular destination \( Y \) is in the \( i \)-th cell. Some of the analysis in the chapter will examine limit behaviors with respect to a sequence of increasingly fine partitions. In those cases, we will often rely on the following construction: Without loss of generality, we assume that the environment \( \mathcal{W} \subset \mathbb{R}^d \) is a hyper-cube with side-length \( L \). For some integer \( r \geq 1 \), we construct a partition \( \mathcal{C}_r \) of \( \mathcal{W} \) by slicing the hyper-cube into a grid of \( r^d \) smaller cubes, each length \( L/r \) on a side; inclusion of subscript \( r \) in our notation will make the construction explicit. The ordering of cells in \( \mathcal{C}_r \) is arbitrary.
Our first result bounds the average match length \( n^{-1}M^*(n) \) in the optimal bipartite matching, asymptotically from below. In preparation for this result we present Problem 5.1, a linear optimization problem whose solution maps partitions to real numbers.

**Problem 5.1 (Optimistic “rebalancing”).**

Minimize \( \sum_{i,j \geq 0} t_{ij} \min_{y \in C_i, x \in C_j} \|x - y\| \)

subject to \( \sum_j t_{ij} = f_Y(C_i) \) for all \( C_i \in \mathcal{C} \),

\( \sum_i t_{ij} = f_X(C_j) \) for all \( C_j \in \mathcal{C} \).

We denote by \( \mathcal{F}(\mathcal{C}; f_Y, f_X) \) the feasible set of Problem 5.1, and we refer to a feasible solution \( T = \{t_{ij}\} \) as a transportation matrix. We will denote by \( \mathcal{W}_C \) the cost of the optimal solution, for reasons which will soon becomes apparent.

**Lemma 5.7 (Lower bound(s) on \( M^* \) in \( \mathbb{R}^{d \geq 1} \)).** For any compact, \( d \geq 1 \) dimensional, Euclidean workspace \( \mathcal{W} \), any finite, measurable partition \( \mathcal{C} \) of \( \mathcal{W} \), and any pair of sequences \( \{X_k\} \) and \( \{Y_k\} \), of i.i.d. random points in \( \mathcal{W} \) (distributed according to \( f_X \) and \( f_Y \), respectively), the optimal bipartite matching cost \( M^*(n) := \min_{\sigma \in \Pi_n} \sum_{k=1}^n \|X_{\sigma(k)} - Y_k\| \) has limit behavior \( \lim \inf_{n \to +\infty} n^{-1}M^*(n) \geq \mathcal{W}_C \) almost surely.

Proof. See Appendix B.2 for the proof.

Naturally, we are interested in the tightest lower bound possible. It turns out the supremum lower bound \( \sup_C \mathcal{W}_C \) is precisely the distance \( \mathcal{W}(f_Y, f_X) \) (the Earth Mover’s distance) between \( f_Y \) and \( f_X \); or, from equation 2.9,

\[
\mathcal{W}(f_Y, f_X) = \inf_{\gamma \in \Gamma(f_Y, f_X)} \int_{\mathcal{W}} \|x - y\| \, d\gamma(y, x). 
\]

(5.2)

Therefore, we can refine Lemma 5.7 as follows.

**Lemma 5.8 (A tight lower bound on \( M^* \) in \( \mathbb{R}^{d \geq 1} \)).** For any compact, \( d \geq 1 \) dimensional, Euclidean workspace \( \mathcal{W} \), and any pair of sequences \( \{X_k\} \) and \( \{Y_k\} \), of i.i.d. random points in \( \mathcal{W} \) (distributed according to \( f_X \) and \( f_Y \), respectively), the optimal bipartite matching cost \( M^*(n) := \min_{\sigma \in \Pi_n} \sum_{k=1}^n \|X_{\sigma(k)} - Y_k\| \) has limit behavior

\[
\lim \inf_{n \to +\infty} n^{-1}M^*(n) \geq \mathcal{W}(f_Y, f_X),
\]

(5.3)

almost surely.

Proof Sketch. The lemma is proved by showing that \( \sup_C \mathcal{W}_C(\mathcal{C}) = \mathcal{W}(f_Y, f_X) \). Intuitively, Problem 5.1 is a discrete approximation and lower bound of (5.2), e.g., it
can be shown that \( \lim_{r \to +\infty} \mathcal{W}_{C_r} - \mathcal{W}(f_Y, f_X) \to 0^- \), where \( C_r \) is the grid partition of \( \mathcal{W} \) into \( d' \) cubes. Applying Lemma 5.7 to this sequence of partitions obtains the lemma. A complete proof of the lemma appears in Appendix B.2, deriving both the approximation bound and the limit of the sequence.

This connection to the Wasserstein distance yields the following notable result.

**Proposition 5.9.** \( n^{-1} M^*(n) \) does not vanish unless \( f_Y = f_X \).

**Proof.** The proposition follows immediately from the fact that the Earth Mover’s distance is known to satisfy the axioms of a metric, e.g., the coincidence axiom, over the space of probability distributions. Nevertheless, we provide a short alternative proof.

Suppose we have distributions \( f_1 \neq f_2 \). Then one can choose \( \epsilon > 0 \) sufficiently small and regions \( A_1 \subseteq A_2 \subseteq \mathcal{W} \), in order to satisfy (i) \( f_1(A_1) > f_2(A_2) \), and (ii) \( \|x - y\| \geq \epsilon \) for all \( x \in A_1 \) and \( y \notin A_2 \). Then

\[
\int_{x,y \in \mathcal{W}} \|x - y\| d\gamma(x, y) \geq \epsilon \gamma \left( \{(x, y) : x \in A_1, y \notin A_2 \} \right) = \epsilon \left[ f_1(A_1) - \gamma \left( \{(x, y) : x \in A_1, y \in A_2 \} \right) \right] \geq \epsilon \left[ f_1(A_1) - f_2(A_2) \right],
\]

uniformly over all \( \gamma \in \Gamma(f_1, f_2) \), which is strictly positive by construction.

The intuition behind this result is that if some fixed region \( A \) in the workspace has unequal proportions of one type of point versus the other, then a positive fraction of the matches associated with \( A \) (a positive fraction of all matches) must have endpoints **outside** of \( A \), i.e., at positive distance. Such an area can be identified whenever \( f_2 \neq f_1 \).

The implication of Lemma 5.8 is that matches have average length asymptotically no less than some constant which depends entirely on the workspace geometry and the spatial distributions of origin and destination points; that constant is generally **non-zero**. The result also has immediate implications for the lengths of stacker crane tours: Recalling the definition of the Mover’s complexity in Section 3.4, let

\[
\mathcal{M}(f) := \mathbb{E}_f \|Y - X\| + \mathcal{W}(f_Y, f_X).
\]

**Theorem 3.2 (Lower bound on SCP* in \( \mathbb{R}^{d \geq 1} \)).** For any compact, \( d \geq 1 \) dimensional Euclidean workspace \( \mathcal{W} \) and any sequence of independent random O/D pairs \( \{(X_k, Y_k)\} \), with distribution \( f \) over \( \mathcal{W}^2 \), the length of the optimal stacker crane tour through the first \( n \) O/D pairs (denoted \( \text{SCP}^*(n; f) \)) has limit behavior

\[
\liminf_{n \to +\infty} n^{-1} \text{SCP}^*_f(n) \geq \mathcal{M}(f) \quad \text{almost surely.} \quad (5.4)
\]
Proof. Recalling equation (5.1), one can write

\[ n^{-1} \text{SCP}^*_f(n) \geq \frac{1}{n} \sum_{k=1}^{n} \|Y_k - X_k\| + n^{-1} \text{M}^*_f(n). \]

By the Strong Law of Large Numbers, the first term of the last expression goes to \( \mathbb{E}_f \|Y - X\| \) almost surely. By Lemma 5.8, the second term is bounded below asymptotically, almost surely, by \( \mathcal{W}(f_Y, f_X) \).

Non-asymptotic Bounds for Matching Costs

Although the almost sure asymptotic bounds are most useful for the objectives of the chapter, in fact, the Earth Mover’s distance bounds the expected match cost from below for all \( n \), non-asymptotically.

Proposition 5.10. \( n^{-1} \mathbb{E}[M^*(n; f_X, f_Y)] \geq \mathcal{W}(f_X, f_Y) \) for all \( n \geq 1 \).

Proof. Let \( f_X \) denote the empirical measure for the set of origins \( \mathcal{X}_n = \{X_k\}_{k=1}^n \), i.e., \( f_X(A) = \frac{|\mathcal{X}_n \cap A|}{n} \) for any measurable area \( A \subset \mathcal{W} \); let \( f_Y \) denote the empirical measure for the set of destinations \( \mathcal{Y}_n = \{Y_k\}_{k=1}^n \). It is known that \( M^*(\mathcal{X}_n, \mathcal{Y}_n)/n = \mathcal{W}(f_X, f_Y) \). The EMD is positive homogeneous in the sense that \( \mathcal{W}(\alpha \mu, \alpha \nu) = \alpha \mathcal{W}(\mu, \nu) \) for measures \( \mu \) and \( \nu \) such that \( |\mu| = |\nu| \) and \( \alpha \geq 0 \). \( \mathcal{W} \) also satisfies the triangle inequality in the sense that \( \mathcal{W}(\mu_1 + \mu_2, \nu_1 + \nu_2) \leq \mathcal{W}(\mu_1, \nu_1) + \mathcal{W}(\mu_2, \nu_2) \), provided that \( |\mu_1| = |\nu_1| \) and \( |\mu_2| = |\nu_2| \). (To see this, observe that the superposition of solutions to the right-hand side (i.e., \( \gamma_1 + \gamma_2 \) is a feasible assignment for the left-hand side.) The trivial combination of positive homogeneity and triangle inequality implies that \( \mathcal{W} \) is a convex function in the measure-pair subspace where \( |\mu| = |\nu| \). Thus, applying Jensen’s inequality, we obtain \( \mathbb{E}_f \mathcal{W}(f_Y, f_X) \geq \mathcal{W}(\mathbb{E}_f f_Y, \mathbb{E}_f f_X) = \mathcal{W}(f_Y, f_X) \). \qed

5.5.2 Upper Bounds on Lengths of Euclidean SCP Tours

In this section we derive an asymptotic upper bound on the length of the optimal stacker crane tour through a set of i.i.d. random O/D pairs \( \{(X_k, Y_k)\} \). The bound is linearly tight in the sense that its rate of growth matches the linear scaling of the lower bound (5.4). In the process of deriving the bound, we will derive a closely related bounds on the total cost \( M^*(n) \) of the optimal bipartite matching between two distinctly i.i.d. random point sets, e.g., between the sets \( \mathcal{Y}_n = \{Y_k\}_{k=1}^n \) and \( \mathcal{X}_n = \{X_k\}_{k=1}^n \) in the case \( Y_k \perp \!\!\!\perp X_k \). We also place particular emphasis on the important case that \( f_Y \neq f_X \).

Our matching bounds will rely on the performance of a hypothetical, randomized algorithm for the stochastic EBMP; (The “hypothetical” algorithm is presented for the sake of analysis only.) The idea of the algorithm is to sample a “shadow mapping” \( \mathcal{X}'_n : \mathcal{Y}_n \to \mathcal{W} \), with two desired properties:

1. First, the average distance \( n^{-1} \sum_{k=1}^{n} \|\mathcal{X}'_n(Y_k) - Y_k\| \), i.e., from a destination \( Y \in \mathcal{Y}_n \) to its “shadow origin” \( \mathcal{X}'_n(Y) \), goes to \( \mathcal{W}(f_Y, f_X) \) as \( n \to +\infty \);
Figure 5-2: Algorithm 2: Demands are labeled with integers. Origin and destination sites are represented by solid and dashed circles, respectively. Origin-to-destination links are shown as black arrows. Shadow origins are shown as dashed squares, with undirected links to their generators (destination sites); also shown are optimal matching links between shadows and origins. Dashed arrows show the resulting induced matching. Note, this solution produces two disconnected subtours \((1, 2, 3)\) and \((4)\).

2. Second, the shadow origins \(X'_n(Y_n) = \{X'_n(Y) : Y \in Y_n\}\) are i.i.d. and independent of the actual origins, but with the same distribution, so that we can apply either Theorem 2.14 or 2.15 to a matching between them.

These properties will be useful for obtaining upper bounds on the cost of the induced matching \(\{(Y, X) : (X'_n(Y), X) \in M^*(X'_n(Y_n), X_n)\}\); such bounds will hold also for the optimal matching.

Algorithm 2 is an essential part of the desired algorithm: Each destination \(y \in Y\) randomly generates an associated shadow origin \(X',\) according to a transportation matrix \(T.\) In line 7, the conditional distribution is defined as

\[
f_X(A | X \in C^j) = \frac{f_X(A \cap C^j)}{f_X(C^j)}. \tag{5.5}
\]

An optimal matching is produced between \(X'\) and \(X,\) which assists in the matching between \(Y\) and \(X;\) specifically, if \(x \in X\) is the point matched to \(X'(y) \in X',\) then the matching produced by Algorithm 2 contains \((y, x).\) An illustrative diagram can be found in Figure 5-2.

Algorithm 2 is specifically designed to have two important properties for random inputs: First, we can characterize \(M^*(X', X)\) using known bounds; e.g., we can show that \(n^{-1}M^*(X', X) \to 0^+\) as \(n \to +\infty.\) Second, \(E \|X'(Y) - Y\|\) is predictably controlled with “tuning” inputs: a partition \(C\) of the environment and matrix \(T.\) Later we will show that \(C\) and \(T\) can be chosen, as a function of \(n,\) so that \(E \|X'(Y) - Y\| \to \mathcal{W}(f_Y, f_X)\) as \(n \to +\infty,\) leading to a bipartite matching algorithm whose performance matches the lower bound of (5.3).

We present the first two properties as formal lemmas:

**Lemma 5.11 (Distribution of \(X').** Given any compact Euclidean workspace \(W\) in dimension \(d \geq 1,\) and

90
Algorithm 2 Randomized EBMP (parameterized)

**Input:** destinations $\mathcal{Y}$ and origins $\mathcal{X}$; ($|\mathcal{Y}| = |\mathcal{X}| = n$)

**Parameters:** distributions $f_Y$ and $f_X$; a partition $C$ of the workspace, and transportation matrix $T$.

**Output:** a bipartite matching between $\mathcal{Y}$ and $\mathcal{X}$.

1. initialize $\mathcal{X}' \leftarrow \emptyset$.
2. initialize matchings $\overline{M} \leftarrow \emptyset$; $\hat{M} \leftarrow \emptyset$; $M \leftarrow \emptyset$.
3. // generate “shadow origins”
4. for $y \in \mathcal{Y}$ do
5. Let $C^y$ be the cell containing destination $y$.
6. Sample $J; J = j$ with probability $t_{ij}/f_Y(C^j)$.
7. Sample $X'$ according to $f_X(\cdot | X \in C^j)$.
8. Insert $X'$ into $\mathcal{X}'$ and $(y, X')$ into $\overline{M}$.
9. end for
10. $\hat{M} \leftarrow$ an optimal bipartite matching between $\mathcal{X}'$ and $\mathcal{X}$.
11. // construct the matching
12. for $x' \in \mathcal{X}'$ do
13. Let $(y, x')$ and $(x', X)$ be the matches in $\overline{M}$ and $\hat{M}$, respectively, whose $\mathcal{X}'$-endpoints are $x'$.
15. end for
16. return $M$

1. any pair of sequences $\{X_k\}$ and $\{Y_k\}$, of i.i.d. random points in $\mathcal{W}$ (distributed according to $f_X$ and $f_Y$, respectively),
2. a finite, measurable partition $C$ of $\mathcal{W}$, and
3. a matrix $T \in \mathcal{T}(C; f_Y, f_X),$

the points $\mathcal{X}'_n$, generated by running Algorithm 2 on $\mathcal{X}_n$ and $\mathcal{Y}_n$ (the first $n$ points in each sequence), with additional inputs $C$ and $T$, are (i) i.i.d., distributed according to $f_X$, and (ii) independent of $\mathcal{X}_n$.

**Proof.** Lemma 5.11 relies on basic laws of probability, and its proof is relegated to Appendix B.3.

The importance of this lemma is that it allows us to apply either Theorem 2.14 or 2.15 of Section 2.2.4 to characterize $M^*(\mathcal{X}', \mathcal{X})$.

**Lemma 5.12 (Cost of the Shadow Map).** For any compact, $d \geq 1$ dimensional Euclidean workspace $\mathcal{W}$, and

1. a finite, measurable partition $C$ of $\mathcal{W}$,
2. a matrix $T \in \mathcal{T}(C; f_Y, f_X)$, and

91
3. a random point $Y$ with distribution $f_Y$ over $W$, 

the expected distance $E_T ||X'(Y) - Y||$ from $Y$ to the shadow point generated by lines 5-7 of Algorithm 2, with inputs $C$ and $T$, is bounded by

$$E_T ||X'(Y) - Y|| \leq \sum_{ij} t_{ij} \max_{y \in C^i, x \in C^j} ||x - y||. \quad (5.6)$$

Proof. Again, see Appendix B.3.

Given a finite partition $C$, it should be desirable to choose $T$ in order to optimize the performance of Algorithm 2; that is, minimize the expected length of the matching produced. We can minimize at least the bound of (5.6) using the solution of Problem 5.2 (shown below).

Problem 5.2 (Pessimistic “rebalancing”).

\[
\begin{align*}
&\text{Minimize } \sum_{i,j} t_{ij} \max_{y \in C^i, x \in C^j} ||x - y|| \\
&\text{subject to } \sum_j t_{ij} = f_Y(C^i) \text{ for all } C^i \in C, \\
&\quad \sum_i t_{ij} = f_X(C^j) \text{ for all } C^j \in C.
\end{align*}
\]

Now we present Algorithm 3, which computes a specific partition $C$, and then invokes Algorithm 2 with inputs $C$ and an optimal transportation matrix $T^*$ according to Problem 5.2:

**Algorithm 3 Randomized EBMP**

**Input:** Destination points $\mathcal{Y} = \{y_1, \ldots, y_n\}$ and origin points $\mathcal{X} = \{x_1, \ldots, x_n\}$

**Parameters:** distributions $f_Y$ and $f_X$

**Output:** a bipartite matching between $\mathcal{Y}$ and $\mathcal{X}$.

**Require:** an arbitrary resolution function $\text{res}(n) \in \omega(n^{1/d})$, where $d$ is the dimension of the space.

1. $r \leftarrow \text{res}(n)$.
2. $C \leftarrow$ grid partition $C_r$, of $r^d$ cubes.
3. $T \leftarrow T^*$, an optimal solution of Problem 5.2.
4. Run Algorithm 2 on $\mathcal{Y}$ and $\mathcal{X}$, with parameters $f_Y$, $f_X$, $C$ and $T^*$, producing matching $M$.
5. return $M$

**Lemma 5.13 (Granularity of Algorithm 3).** For any compact, $d \geq 1$ dimensional Euclidean workspace $\mathcal{W}$, and any probability distributions $f_X$ and $f_Y$ over $\mathcal{W}$: the expected distance $E \|X'(Y) - Y\|$ from a random point $Y$, distributed according
to $f_Y$, to the shadow point generated by lines 5-7 of Algorithm 2, running under Algorithm 3 (with parameters $f_Y$ and $f_X$), is bounded by

$$E\|A'(Y) - Y\| - \mathcal{W}(f_Y, f_X) \leq 2r^{-1} \text{diam } \mathcal{W} \sqrt{d},$$

where $r$ is the resolution parameter determined by Algorithm 3.

**Proof.** See Appendix B.3 for the proof. 

We are now in a position to present an upper bound on the total cost of the optimal matching between random sets that holds in the general case when $f_Y \neq f_X$.

**Lemma 5.14 (Upper bound on $M^*$ in $\mathbb{R}^{d \geq 3}$).** For any compact, $d \geq 3$ dimensional Euclidean workspace $\mathcal{W}$, and any pair of sequences $\{X_k\}$ and $\{Y_k\}$, of i.i.d. random points in $\mathcal{W}$ (distributed according to $f_X$ and $f_Y$, respectively), there is a constant

$$\alpha_M \leq \beta_{M,d} \min_{\varphi \in \Phi_{f_X,f_Y}} \left\{ \int_{\mathcal{W}} \varphi(x)^{1-1/d} \, dx \right\}$$

such that the optimal bipartite matching cost has limiting behaviour

$$\limsup_{n \to +\infty} \frac{M^*(n; f) - n\mathcal{W}(f_Y, f_X)}{n^{1-1/d}} \leq \alpha_M \quad \text{almost surely.}$$

**Proof.** The proof relies on a characterization of the total cost of the matching produced by Algorithm 3 ($A_3$), which also bounds the length of the *optimal* matching. By the triangle inequality, the cost $A_3(Y, X)$ of its matching is no larger than the sum of two terms: (i) the sum $\sum_{k=1}^{n} \|X_k' - Y_k\|$, of distances from the points in $Y$ to their shadows; we can write this sum as $M^1(Y, X')$, denoting the “identity” matching between the two ordered sets, i.e., $\{(Y_k, X'_k) : k = 1, \ldots, n\}$; plus (ii) the total cost of the optimal matching $M^*(X', X)$ (also in the Euclidean distance sense) between the shadow origins and the actual ones. Thus, we have

$$A_3(Y, X) \leq M^1(Y, X') + M^*(X', X).$$

By subtracting on both sides of equation (5.9) the term $n\mathcal{W}(f_Y, f_X)$, and dividing by $n^{1-1/d}$, we obtain

$$\frac{A_3(Y, X) - n\mathcal{W}(f_Y, f_X)}{n^{1-1/d}} \leq \frac{M^1(Y, X') - n\mathcal{W}(f_Y, f_X)}{n^{1-1/d}} + \frac{M^*(X', X)}{n^{1-1/d}}$$

$$\leq \frac{M^1(Y, X')}{n^{1-1/d}} + \frac{M^*(X', X)}{n^{1-1/d}} + O\left(\frac{n^{1/d}}{r}\right),$$

where the last equality follows from Lemma 5.13. We observe that $n \mathbb{E} \|A'(Y) - Y\|$ is the expectation of $M^1(Y, X')$, and so the first term goes to zero almost surely (absolute differences law; see Theorem 2.8). The resolution function of Algorithm 3 ensures that the third term vanishes. For $d \geq 3$, Lemma 5.11 allows us to apply
Theorem 2.14 to the second term, obtaining the limit
\[
\lim_{n \to +\infty} \frac{M^*(\mathcal{X}', \mathcal{X})}{n^{1-1/d}} = \beta_{M,d} \int_{\mathcal{W}} \varphi_X(x)^{(d-1)/d} \, dx \quad \text{almost surely.}
\]
Collecting these results, and the sure bound \( M^* \leq \mathcal{A}_3 \), we obtain the inequality in (5.8) with \( \varphi = \varphi_X \) in (5.7).

To complete the proof for the case \( d \geq 3 \), we observe that Algorithm 2 could be defined alternatively as follows: the points in \( \mathcal{X} \) generate a set \( \mathcal{Y}' \) of shadow sites; the intermediate matching is now between \( \mathcal{Y}' \) and \( \mathcal{Y} \). One can then prove results congruent with the results in Lemmas 5.11, 5.12, and 5.13. By following the same line of reasoning, one can finally prove the inequality in (5.8) with \( \varphi = \varphi_Y \) in (5.7).

**Lemma 5.15 (Upper bound on \( M^* \) in \( \mathbb{R}^2 \)).** For any compact, planar workspace \( \mathcal{W} \subset \mathbb{R}^2 \), there is a constant \( K_{\mathcal{W}} \) such that for any pair of sequences \( \{X_k\} \) and \( \{Y_k\} \), of i.i.d. random points in \( \mathcal{W} \) (distributed according to \( f_X \) and \( f_Y \), respectively), the optimal bipartite matching cost has limiting behaviour
\[
\frac{M^*(n; f) - n \mathcal{W}(f_Y, f_X)}{\sqrt{n \log n}} \leq K_{\mathcal{W}}, \tag{5.10}
\]
with high probability \((1 - o(1))\) as \( n \to +\infty \).

**Proof.** The proof for the case \( d = 2 \) follows essentially the same logic, except Theorem 2.15 is used instead of Theorem 2.14, where \( \gamma \) may depend on \( \mathcal{W} \).

**Remark 5.16.** Suppose the points in sets \( \{X_k\} \) and \( \{Y_k\} \) are not independent, but instead they are generated by an i.i.d. random sequence \( \{(X_k, Y_k)\} \), with joint distribution \( f \). Unfortunately, Lemma 5.11 does not hold under this dependence structure, and so Lemmas 5.14 and 5.15 do not follow. However, one way to recapture the necessary independence is to inject another shadow set \( \mathcal{X}'' \), purely i.i.d. with distribution \( f_X \), between \( \mathcal{X}' \) and \( \mathcal{X} \). Thus, a matching can be produced with cost no greater than \( M^1(\mathcal{Y}, \mathcal{X}') + M^*(\mathcal{X}', \mathcal{X}'') + M^*(\mathcal{X}'', \mathcal{X}) \), where Lemma 5.11 holds for the last two matches.

We can leverage the previous results to derive the main result of this section, which is an asymptotic upper bound for the optimal cost of the ESCP. In addition to having the same linear scaling as (5.4), the bound also includes “next-order” terms: Suppose a SPLICE algorithm uses a c-optimal TSP tour through the origins (with shortcutting) to connect subtours. For example, LARGEARCS is SPLICE with the 2-optimal MST-based TSP heuristic.

**Theorem 5.17 (Upper bound on SPLICE in \( \mathbb{R}^{d \geq 3} \)).** Let \( \mathcal{A} \) be a SPLICE+c-TSP algorithm in \( \mathbb{R}^d \), \( d \geq 3 \), for some \( c \geq 1 \). For any compact, dimension \( d \geq 3 \), Euclidean workspace, and any sequence of independent random O/D pairs \( \{(X_k, Y_k)\} \), with distribution \( f \) over \( \mathcal{W}^2 \), there is a constant
\[
\alpha \leq 2 \alpha_M + c \beta_{\text{TSP},d} \int_{\mathcal{W}} \varphi_X^{1-1/d}(x) \, dx \tag{5.11}
\]
(\alpha_M \text{ from (5.7)}) such that
\[
\limsup_{n \to +\infty} \frac{A(n; f) - nM(f)}{n^{(d-1)/d}} \leq \alpha, \quad \text{almost surely.} \tag{5.12}
\]

If \( Y \perp X \), then the result holds for any SPLICE algorithm, with \( \alpha \leq \alpha_M \).

Proof. Let \( X_n \) and \( Y_n \) denote the origin and destination sets, respectively. One can write
\[
A(n) \leq \sum_{k=1}^n \|Y_k - X_k\| + M^*(Y_n, X_n) + c \text{TSP}^*(X_n) \tag{5.13}
\]
\[
= \left( \sum_{k=1}^n \|Y_k - X_k\| - n\mathbb{E}_f \|Y - X\| \right) + \left( M^*(Y_n, X_n) - n\mathbb{W}(f_Y, f_X) \right) + nM(f) + c \text{TSP}^*(X_n). \tag{5.14}
\]

The following results hold almost surely: The first term of the final expression (5.14) is \( o(n^{1-1/d}) \) (absolute differences law); by Lemma 5.14 and Remark 5.16, the second term is bounded by \( 2 \alpha_M n^{1-1/d} + o(n^{1-1/d}) \); finally, by BHH (Theorem 2.11), the TSP term is bounded by \( (c \beta_{\text{TSP},d} \int \frac{1}{\mathcal{W}_X} x^{1-1/d}(x) dx) n^{1-1/d} + o(n^{1-1/d}) \). Thus, the claim is obtained by moving the second-to-last term (Mover's complexity) to the other side of the inequality, dividing on both sides by \( n^{1-1/d} \), and taking the limit as \( n \to +\infty \).

Finally, if \( Y \perp X \), then (5.13) will hold for any SPLICE algorithm, except we need not applying Remark 5.16, and we can replace the TSP term with the maximum extra length incurred by subtour connection, e.g., \( N_n \mathcal{W} \), where \( N_n \) is the number of generated subtours. By Remark 5.6, one has \( \lim_{n \to +\infty} N_n/n^{1-1/d} = 0 \) almost surely. \( \square \)

**Theorem 5.18 (Upper bound on SPLICE in \( \mathbb{R}^2 \)).** Let \( A \) be a SPLICE +c-TSP algorithm in \( \mathbb{R}^2 \). For any compact, planar workspace \( \mathcal{W} \subset \mathbb{R}^2 \) and any sequence of independent random O/D pairs \( \{(X_k, Y_k)\} \), with general distribution,
\[
\frac{A(n; f) - nM(f)}{\sqrt{n \log n}} \leq K_W \tag{5.15}
\]
with high probability \( (1 - o(1)) \) as \( n \to +\infty \). \( (K_W \text{ from Lemma 5.15}) \). If \( Y \perp X \), then the result holds for any SPLICE algorithm.

Proof. The proof is almost identical to the proof of Theorem 5.17. However, the result is ultimately weaker, because the convergence of the second (matching) term in (5.14) is only with high probability for \( d = 2 \) (Lemma 5.15). (Almost sure convergence for the other terms implies convergence in probability.) Moreover, since the TSP term is \( O(n^{1/2}) \) for \( d = 2 \), it is technically dominated by the matching term, which is \( O((n \log n)^{1/2}) \). \( \square \)
Practically speaking, it is quite hard to justify neglecting a term that is dominated by a mere factor of $(\log n)^{1/2}$, e.g., the length of the connecting TSP tour versus the cost of the matching. However, on large instances, short-cutting procedures tend to eliminate the TSP tour almost entirely, so it is already insignificant on large instances.

**Remark 5.19.** Theorems 5.17 and 5.18 hold trivially for the optimal stacker crane tour, because of the obvious sure bound $\text{SCP}^* \leq A$ for any algorithm $A$.

### 5.6 Simulation Results

In this section, we present simulation results about (i) the performance of SPLICE, by way of the LARGEARCS algorithm (including a comparison with SMALLARCS), and (ii) the rate of growth of the cost of the optimal bipartite matching between sets of random points.

#### 5.6.1 Performance of SPLICE

We begin the simulation section with a discussion of the performance of SPLICE. We choose the LARGEARCS algorithm as our champion, since it has been state-of-the-art (as part of FHK [57]) since 1976. We examine the convergence of the cost of its solution to that of the optimal solution, and we compare the runtime of LARGEARCS with that of an exact algorithm.

The optimal bipartite matchings used in LARGEARCS (and line 1 of any SPLICE algorithm) were computed with the open-source GNU Linear Programming Toolkit (GLPK) software on a linear program written in MathProg/AMPL; for comparison with LARGEARCS, optimal stacker crane tours were computed using the same software on the integer programming model [80] of a straightforward reduction to the asymmetrical TSP [75]. (Simulations were run on a laptop computer with a 2.66 GHz dual core processor and 2 GB of RAM.)

Figure 5-3(a) shows ratios $\text{L.ARC}/\text{SCP}^*$ (lower trend), observed over a set of randomly generated samples (twenty-five (25) trials in each of seven (7) size categories); here, L.ARC denotes the matching [total cost] produced by LARGEARCS. Points were sampled i.i.d., uniformly over the unit cube $[0, 1]^3$. Also shown are the ratios $\text{SM.ARC}/\text{SCP}^*$ (upper trend) on the same samples, where SM.ARC denotes the matching produced by SMALLARCS. As predicted in Section 5.4.2, SMALLARCS remains distinctly suboptimal (for $n \geq 20$ it never wins), realizing a range of factors; e.g., for $n$ between 30 and 100, it is generally within 1.2–1.6 times optimal. In contrast, one can see that the convergence of the factor of LARGEARCS to $1^+$ is extremely fast. (In other simulations (not shown), we sampled from other distributions—in particular those of Section 3.9—but the choice had little effect on performance.)

Figure 5-3(b) shows the ratios $\text{T.EXACT}/\text{T.LARC}$ for the same set of problem instances, where $\text{T.EXACT}$ is the runtime of our exact algorithm and $\text{T.LARC}$ is the runtime of LARGEARCS. In the range shown, the exact algorithm involves factor approximately $n$ extra runtime versus LARGEARCS. For numbers larger than $n \approx 100$...
5.6.2 Cost Bounds—First- and Next-Order Asymptotics

Next, we examine the costs of random matchings and compare empirical results with the bounds predicted by Lemmas 5.8 and 5.14 (in terms of cardinality). We focus our attention on the two distributions from Section 3.9.2 for which \( \mathcal{W} \neq 0 \); specifically, Cases III and IV, reproduced below.

**Case III Unit Cube Arrangement:** The distribution of origin sites \( f_X \) places one-half of its probability uniformly over a unit cube centered along the \( x \)-axis at \( x = -4 \), and the other half uniformly over the unit cube centered at \( x = -2 \). The distribution of destination sites \( f_Y \) places one-half of its probability uniformly over the cube at \( x = -4 \) and the other half over a new unit cube centered at \( x = 2 \).

**Case IV Co-centric Sphere Arrangement:** Origins are uniformly distributed over a sphere of radius \( R = 2 \), and destinations are uniformly distributed over a sphere of radius \( r = 1 \). Both spheres are centered at the origin.

For the purpose of illustration, the samples from Figures 3-2(a) and 3-2(b) are reproduced in Figures 5-4(a) and 5-4(b), respectively, next to optimal bipartite matchings drawn for each sample set.

Experimental results are shown in Figure 5-5. Figure 5-5(a) (top) shows a scatter plot of \((n, n^{-1}M^*)\) for Case III, with one point for each of twenty-five (25) trials, in
Figure 5-4: Illustration of bipartite matching of samples from non-uniform distributions (Cases III and IV): $n = 100$ samples for each distribution; origin sites are shown as (red) triangle markers; destination sites are shown as (blue) circles. Plots on the left show samples alone; plots on the right include dashed lines between matched points in the optimal matching.
Figure 5-5: Scatter plots of \((n, n^{-1}M^*)\) (top) and \((n, n^{-2/3}(M^* - nW))\) (bottom), with one point for each of twenty-five trials per size category, for Cases III and IV.

Each of seven (7) size categories; i.e., the x-axis denotes the number of points in each set, and the y-axes denote the average length of a match in the optimal bipartite matching. Additionally, the plot shows a curve (solid line) through the empirical mean in each size category, and a dashed line showing the Earth Mover’s distance \(W(f_Y, f_X)\), the predicted asymptotic limit of the sequence \(n^{-1}M^*(n)\). Figure 5-5(b) (top) is analogous to Figure 5-5(a) (top), but for random samples under Case IV. Both plots exhibit the predicted approach of \(n^{-1}M(n)\) to the Earth Mover’s distance, but the convergence in Figure 5-5(b) (top) appears slower because \(W\) is smaller.

The bottom plots of Figure 5-5 are scatter plots of \((n, n^{-2/3}(M^* - nW))\) from the same data, with another solid curve through the empirical mean. Also shown in the plots are two dashed, horizontal lines. The top line in each case is the level of the constant \(\alpha_M\) of (5.7), called \(\alpha\) in Table 5.1, below. Both plots indicate asymptotic convergence (at least in the average) to a constant no larger than \(\alpha_M\).

<table>
<thead>
<tr>
<th>Case III</th>
<th>(|E_f|Y - X|)</th>
<th>(W)</th>
<th>(M)</th>
<th>(\alpha/\beta_{M,3})</th>
<th>(\alpha^3/M^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case III</td>
<td>(\approx 3.20)</td>
<td>(2)</td>
<td>(\approx 5.20)</td>
<td>(\sqrt[3]{2})</td>
<td>(\approx 2.62 \times 10^{-2})</td>
</tr>
<tr>
<td>Case IV</td>
<td>(\approx 1.66)</td>
<td>(0.75)</td>
<td>(\approx 2.41)</td>
<td>(\sqrt[3]{4\pi/3})</td>
<td>(\approx 2.59 \times 10^{-1})</td>
</tr>
</tbody>
</table>

Table 5.1: Selected rows from Table 3.1. Compiled statistics for simulated demand models, with estimates of \(E_f\|Y - X\|\), \(W\), \(M\), and \(\alpha\).

### 5.7 Conclusion

In this chapter we have presented new analysis of the popular LARGEARCS heuristic for the Stacker Crane problem, under a general probabilistic framework, and demonstrated that it is asymptotically optimal, almost surely. Moreover, we have embedded LARGEARCS in a class SPLICE of polynomial-time algorithms for the stochastic (Euclidean) SCP, which are all asymptotically optimal under the condition \(Y \perp \perp X\).
(We believe optimality holds even in the absence of this condition.) More importantly, we characterized analytically the length of LARGEARCS, SPLICE, and optimal SCP tours, deriving bounds that had been crucial to the analysis of the 1-DPDP model in Chapter 3. While it has been obvious that the optimal SCP tour should grow in length linearly in \( n \), we have characterized the rate of growth explicitly, revealing a previously-hidden dependence on a derivable statistic, connected to the study of mass transportation theory. We hope that the techniques introduced in the chapter (e.g., coupling the EBMP with the theory of random permutations) may come to bear in analysis and heuristics for other hard combinatorial problems.

This study leaves numerous important extensions open for further research. First, we are interested in precisely characterizing the convergence rate of LARGEARCS and SPLICE to optimal solutions. Second, we are interested in mitigating the violent disconnect (e.g., in the right-hand side constant in (5.11)) between the cases \( Y \parallel X \) and \( Y \perp X \). Third, while our model assumes omnidirectional service vehicle (i.e., sharp turns are allowed), we hope to develop approximation algorithms for the SCP where the vehicle has differential motion constraints (e.g., bounded curvature), as is typical, for example, with unmanned aerial vehicles, or “street constraints”, e.g., for vehicles on urban streets, or between rows of shelves in a warehouse.

Sources: The SPLICE algorithm class was proposed in [123], although it had already been common in practice to use optimal bipartite matchings between the sets of origin points and destination points to construct a set of stacker crane tours, servicing all demands. However, [123] first proved that the SPLICE algorithms generate \( \Theta(\log(n)) \) subtours for the case \( Y \perp X \), but otherwise generally distributed, and thus proved the asymptotic optimality of the class. The asymptotic bounds for the stochastic Euclidean bipartite matching and stacker crane functionals with distinct first and second distributions were first presented in [124]. To the author’s best knowledge, they were the first such results. ([125] summarizes [93, 123, 124] in the form of a journal paper.)
Part II
Transportation Problems on Road Networks
Chapter 6

Static and Dynamic Taxiing with Random Demands on Road Networks

6.1 Introduction

With this chapter we begin Part II of the thesis, which takes a deeper look at some of the practical issues concerning shared-vehicle, one-way transportation systems. The common theme of this study will be a shift of focus away from free travel in the Euclidean plane, and toward a model of road network environments for capturing the “street constraints” common to most transportation scenarios. There are several compelling reasons to focus on environments where street constraints are in play:

First and foremost, they provide a more convincing model of the urban or otherwise manicured environments in which problems involving physical transport tend to be set. Euclidean models are often used in basic research studies for sake of simplicity. Fortunately for previous studies, e.g., of the DTRP with UAVs in a large airspace, the Euclidean setting already captured practical motion constraints. Physical transport, on the other hand, is usually carried out by ground vehicles on a system of paths. It has been observed that the true ground distance between two points on a road network can be poorly and unpredictably approximated by Euclidean, or bird’s-eye, distance. The plot from [84] in Figure 6-1, of the ratio of such distances observed between random points on a map of Tokyo, provides some evidence of this phenomenon, especially at short range. Such inaccuracies can pose a significant challenge when trying to obtain important statistical quantities with reasonable precision, e.g., the Earth Mover’s distance. This can be quite unfortunate, since, for example, Chapter 3 demonstrated that the equations governing the performance of taxi routing policies can be extremely sensitive to the EMD through the Mover’s complexity.

In addition to better model accuracy, the restrictive geometry of road networks allows for significant improvements in the runtime of matching algorithms and related heuristics to solve the SCP. For example, we will present a new efficient matching algorithm on road-maps in Chapter 7, which forms the basis of a SPLICE algorithm
Figure 6-1: Ratios of shortest path (roadmap) distance versus Euclidean (bird’s eye) distance. Image obtained from [84], produced by randomly sampling a map of Tokyo.

Presented in Section 6.7 to simulate Stacker Crane policies. Road networks also admit simplified numerical methods for statistical computations, e.g., to compute the EMD, which we will discuss in great detail in Chapter 8.

Contributions. We begin our study of transportation on road networks in the present chapter, by presenting parallel versions of the models and results in Chapters 3 and 5, but in the setting of road networks instead. We derive new asymptotic cost bounds for optimal matchings and SCP tours under a general probabilistic framework on road networks. Then, we present the 1-DPDP on road networks, derive a necessary and sufficient condition for stabilizability, and provide simple extensions and analysis of the vehicle routing policies from Chapter 3 on road networks. We find that the Earth Mover’s distance plays the same important role as it did in Euclidean settings, and we posit a new, different scaling law for the average system time $T$ as a function of system utilization.

Organization. The chapter is organized as follows. First, in Section 6.2 we develop some necessary mathematical formalism for road network geometry and probability models. Formal treatments of road networks as continuous metric spaces are somewhat rare in literature. (Okabe and Sugihara [84] explore one similar yet distinct geometrical approach.) We present a preliminary discussion of bipartite matching on a segment in Section 6.3, and then in Section 6.4 we derive extensions—to the road network setting—of the probabilistic cost bounds for optimal bipartite matchings of Chapter 5. In Section 6.5 we extend such matching bounds to derive probabilistic cost bounds for optimal stacker crane tours through large random sets of demands on roadmaps. Using the new probabilistic bounds, we consider the 1-DPDP in the road network setting in Section 6.6: We re-derive performance bounds for the policies from Chapter 3, re-derive policy-independent lower bounds, and we re-derive the necessary and sufficient stability condition, all for the road network setting. In Section 6.7 we present a new simulation study, paralleling the simulations of Section 3.9. Finally, in Section 6.8, we offer concluding remarks.
6.2 Road Network Environments

At its simplest, a road network is a set of lines or curves connected together into a particular pattern by their endpoints. The distance between points on a roadmap is the minimum distance by which a particle (or vehicle) could reach one point from the other while constrained to travel on the curves, or roads. Such curves by their nature are usually characterized by some higher-dimensional spatial embedding, e.g., they may be drawn on a map (the plane). However, we can describe the geometry of a road network without regard to such embedding.

6.2.1 Notation and Representation

Unlike the formidable and versatile Euclidean plane ($\mathbb{R}^2$), there are many geometrically distinct road networks. Thus, we are in fact discussing a class of metric spaces. Every road network, however, is a continuous, one-dimensional space, with a notion of distance which includes but is not limited to the traditional notion of distance between vertices in a graph. In order to describe a particular road network, we require a succinct system for representation. In this section, we introduce the notation of a concise mathematical description for the class of road networks using a graph theoretical framework. Despite the mathematical formalism, the reader should bear in mind that we will put into formal terms the most natural understanding of road network geometry.

It is common practice to represent the topology of a roadmap using an undirected, weighted graph or multi-graph $(\mathcal{V}, \mathcal{R})$, possibly with loops, where the edges $\mathcal{R}$ correspond to roads in the roadmap and are labeled with lengths; the vertices $\mathcal{V}$ support the interconnections of the roads. Then, one can attach to such graph a coordinate system, first fixing an orientation of the network.

**Definition 6.1 (Orientation).** An orientation of an undirected graph $G$ is the assignment of a direction to each edge in $G$, resulting in a directed graph.

Given a fixed orientation of the roadmap graph, every point on the roadmap continuum can be described unambiguously by a tuple, or address $(r, y)$, of a road $r \in \mathcal{R}$ and a real-valued coordinate $y \in [0, L_r]$, where $L_r$ is the length of road $r$. There is an intuitive notion of “roadmap distance” between points described by such addresses, arising from two basic assertions: (i) there is a path between any two points on the same road, of length equal to the difference between their address coordinates; (ii) there is a special point for every roadmap vertex $u \in \mathcal{V}$ which is at the respective endpoints of all the roads adjacent to $u$, simultaneously. The distance between two points then is the length of the shortest path between them. Throughout our discussions, we assume an orientation of the road system has been fixed, so that $\mathcal{R}$ is directed.

A set $\mathcal{R}_d \subseteq \mathcal{R}$ may be used to specify those roads in $\mathcal{R}$ which are defined to be “one-way”, a distinction which we will discuss later. However, unless otherwise stated, in the thesis we will always assume that $\mathcal{R}_d = \emptyset$, i.e., none of the roads are thusly marked. As a final remark, it is important to recognize that a physical road
network may have multiple distinct roads going between two particular interchanges. Therefore, the di-graph \((V, R)\) may well be a multi-graph. As a result, we will rarely use the endpoint-tuple notation commonly used to denote edges.

### 6.2.2 The Geometry of Road Networks

A road network is described by a road map.

**Definition 6.2 (RoadMap).** A road map is the collection \(((V, R), L, R_d)\) of an oriented (directed) graph \((V, R)\)—a potentially loopy, multi-digraph—with a road length mapping \(L : R \to \mathbb{R}_{\geq 0}\) and a set \(R_d \subseteq R\) of “one-way” roads.

Consider a roadmap \(((V, R), L, R_d)\). For each \(r \in R\), let \((r, \mathcal{B}_r)\) denote the metric space which is isomorphic to the interval \([0, L_r] \subset \mathbb{R}\) through the bijective mapping

\[
y \mapsto \begin{cases} 
  r^- & y = 0, \\
  (r, y) & y \in (0, L_r), \\
  r^+ & y = L_r.
\end{cases}
\]

This space represents a restriction of the road network to only those points on the road \(r\). Topological notions such as interior (int), closure (cl), and boundaries (bd) of subsets of \(r\), and even the Borel sets \(\mathcal{B}_r\), are obtained trivially through this isomorphism. We will refer to the image \(I_r\) of any interval \(I \subset [0, L_r]\) as a road interval. Note that \([0, L_r]_r = r, (0, L_r)_r = \text{int}(r)\), etc. On \(\text{int}(r)\), for each \(r \notin R_d\), the metric \(D_r\) is simply

\[
D_r(p, p') = |y' - y| : p = (r, y), p' = (r, y').
\]

The fairly trivial extension of \(D_r\) to the whole road, i.e., including \(\text{bd}(r) = \{r^-, r^+\}\), is given by Table 6.1.

If \(r \in R_d\), then on \(\text{int}(r)\) \(D_r\) has behavior

\[
D_r(p, p') = [y' - y]_+ : p = (r, y), p' = (r, y'),
\]

where

\[
[x]_+ = \begin{cases} 
  x & \text{if } x \geq 0, \\
  +\infty & \text{otherwise}.
\end{cases}
\]

![Figure 6-2: The simplest geometric network, consisting of a single road.](image-url)
<table>
<thead>
<tr>
<th>$r^-$</th>
<th>$(r, y') : y' \in (0, L_r)$</th>
<th>$r^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r^-$</td>
<td>$0$</td>
<td>$L_r$</td>
</tr>
<tr>
<td>$(r, y) : y \in (0, L_r)$</td>
<td>$y$</td>
<td>$</td>
</tr>
<tr>
<td>$r^+$</td>
<td>$L_r$</td>
<td>$0$</td>
</tr>
</tbody>
</table>

Table 6.1: The single-road metric $\mathcal{D}_r$ on $r \notin R_d$ (i.e., regular road). Choose $p$ from the first column, and choose $p'$ from the first row.

and is given by Table 6.2 on the whole road. In the case that $r$ is a loop road, i.e.,

<table>
<thead>
<tr>
<th>$r^-$</th>
<th>$(r, y') : y' \in (0, y)$</th>
<th>$r^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r^-$</td>
<td>$0$</td>
<td>$L_r$</td>
</tr>
<tr>
<td>$(r, y) : y \in (0, L_r)$</td>
<td>$+\infty$</td>
<td>$</td>
</tr>
<tr>
<td>$r^+$</td>
<td>$+\infty$</td>
<td>$+\infty$</td>
</tr>
</tbody>
</table>

Table 6.2: The single-road metric $\mathcal{D}_r$ on $r \in R_d$ (i.e., one-way road). Choose $p$ from the first column, and choose $p'$ from the first row.

$r^- = r^+$, then one should ignore the rows and columns corresponding to $r^+$.

**Definition 6.3 (Road network point set).** The point set $\mathcal{R}$ of a road network described by a roadmap $(V, R)$ is precisely the set $\bigcup_{r \in R}[0, L_r]$, i.e.,

$$\mathcal{R} = \{(r, y) : r \in R, 0 < y < L_r\} \cup V.$$  

**Definition 6.4 (Road network union metric).** Let $\tilde{\mathcal{D}}(p, p') = \min_{r \in R} \mathcal{D}_r(p, p')$; here, we use the common convention $\mathcal{D}_r(p, p') = +\infty$ wherever it is not defined. We call $\tilde{\mathcal{D}}$ the union metric although it is not a proper metric, because, e.g., it does not satisfy the triangle inequality.

![Diagram](image)

Figure 6-3: A two-road roadmap, with points $X$, $Y$, and $Z$ demonstrating failure of union metric $\tilde{\mathcal{D}}$ to satisfy triangle inequality.

For example, in Figure 6-3, $\tilde{\mathcal{D}}(X, Z) = \mathcal{D}_{r_1}(X, Z) < +\infty$ and $\tilde{\mathcal{D}}(Z, Y) = \mathcal{D}_{r_2}(Z, Y) < +\infty$, however $\mathcal{D}_r(X, Y)$ is not defined for any $r \in R$, and therefore $\tilde{\mathcal{D}}(X, Y) = +\infty$. Next, we will derive the proper distance metric from the union metric; however, at this point it is intuitively clear that the shortest path distance from $X$ to $Y$ is equal to $\mathcal{D}_{r_1}(X, Z) + \mathcal{D}_{r_2}(Z, Y)$. 

107
The shortest-path distance metric, which encodes formally our intuitive notion of roadmap distances, is precisely the intrinsic metric derived from the union metric by Definition 2.6:

**Definition 6.5 (Shortest-path distance on road networks).** Let $\mathcal{D}_R$ denote the intrinsic metric of $\tilde{\mathcal{D}}$.

That is, for every pair of points $(p_1, p_2) \in \mathcal{R}^2$, the distance $\mathcal{D}_R(p_1, p_2)$ is equal to the length of the shortest path between addresses of $p_1$ and $p_2$, respectively, according to $\tilde{\mathcal{D}}$. Although the union “metric” $\tilde{\mathcal{D}}$ is not a proper metric, the intrinsic metric is one.

**Definition 6.6 (Road Network).** A road network is any metric space $(\mathcal{R}, \mathcal{D}_R)$ generated in this way by some roadmap (representation) $((\mathcal{V}, R), L, R_d)$.

**An Explicit Formulation of $\mathcal{D}_R$**

A key property of the collection $\{\mathcal{D}_r : r \in R\}$ of metrics is that if $P$ is the shortest path on the network from $p \in \mathcal{R}$ to $p' \in \mathcal{R}$, then there exists a sequence of points $\{p_k\}_{k=1}^{n \geq 2}$, $p_k \in \mathcal{R}$, with endpoints $p_1 = p$ and $p_n = p'$, and another sequence of roads $\{r_k\}_{k=1}^{n-1}$, $r_k \in R$, such that

$$|P| = \sum_{k=1}^{n-1} \mathcal{D}_{r_k}(p_k, p_{k+1}) < +\infty. \quad (6.1)$$

In fact, there exists such connecting sequence $\{p_k\}$ where all points besides possibly $p_1$ and $p_n$ are supported entirely on $\mathcal{V}$. The length of the shortest interstitial sojourn can be represented, e.g., by a Dijkstra shortest path (dijkstra) between at most two vertices; thus, the minimum of (6.1) over sequences $(p_k)$ can be reduced to a finite closed form

$$\mathcal{D}_R(p, p') \doteq \min\left\{ \tilde{\mathcal{D}}(p, p'), \min_{r_1, r_2 \in R, \begin{array}{c} u \in \{r_1^{-}, r_1^{+}\}, \hfill \varepsilon \in \{r_2^{-}, r_2^{+}\}\end{array}} \mathcal{D}_{r_1}(p, u) + \text{dijkstra}(u, v) + \mathcal{D}_{r_2}(v, p') \right\}.$$ 

One quite deceptive property of this metric is that the second term of the outer minimum may indeed be chosen even if the first term is finite. For example, consider the distance between points $X$ and $Y$ on the network of Figure 6-4 (drawn to scale).

**6.2.3 A Probability Model for Random Points**

A major focus of the thesis is on properties of combinatorial problems whose instances are randomly-generated sets of points. In order to discuss random points on road networks formally, we must first develop a probabilistic model.
Figure 6-4: A ring network with two roads, \( r_1 \) and \( r_2 \). The shortest path between \( X \) and \( Y \) covers the space to left of \( X \) and \( Y \), thus using both roads at least partially. The union metric describes the length of the longer path on \( r_2 \), using the space to the right of \( X \) and \( Y \).

A crucial property of road networks, although hardly surprisingly, is that as continuous metric spaces, they are complete and separable.

**Proposition 6.7.** Road networks are complete and separable.

*Proof.* The point set of a road network is composed of (i) a set of open intervals (the road interiors), all disjoint, and (ii) another finite point set, i.e. \( V \). The only limit points missing from the collection of roads are the interval boundaries, which are finite in number and are “filled in” by (ii). Injection of the finite set of points \( V \) cannot introduce new limit points, therefore \( \mathcal{R} \) is complete. \( \mathcal{R} \) is separable because it is a finite union of separable components.

This property allows us to pursue a formal probabilistic model using the established machinery of Borel sets and even Lebesgue sets.

Given a road network metric space \( (\mathcal{R}, \mathcal{D}_\mathcal{R}) \), let \( \mathcal{B}_\mathcal{R} \) denote the Borel sets (\( \sigma \)-algebra) generated by all the open sets in the topology defined on \( \mathcal{R} \) by \( \mathcal{D}_\mathcal{R} \). Let \( \mathcal{F}_\mathcal{R} \) denote the corresponding Lebesgue measurable sets.

**Definition 6.8 (Absolute continuity of measure).** A measure \( \mu \) over a measurable roadmap \( (\mathcal{R}, \mathcal{F}_\mathcal{R}) \) is absolutely continuous if there exists a Lebesgue measurable mapping \( \varphi_\mu \) such that \( \mu(A) = \int_A \varphi_\mu(p) \, dp \) for all \( A \in \mathcal{F} \); equivalently, if there exists a set \( \varphi = \{ \varphi_r : \mathcal{R} \to \mathbb{R} \}_{r \in \mathcal{R}} \) of integrable mappings such that

\[
\mu(A) = \sum_{r \in \mathcal{R}} \int_{y : (r,y) \in A} \varphi_r(y) \, dy. \tag{6.2}
\]

We call the components of \( \varphi \) the road densities.

**Definition 6.9 (Cumulative density function).** Given a Lipschitz density function \( \varphi : [0, L] \to \mathbb{R}_{\geq 0} \), let

\[
\Phi(y; \varphi) = \int_0^y \varphi(y') \, dy'.
\]

\( \Phi(\cdot; \varphi) \) is called the cumulative density function (cdf) of \( \varphi \), and for \( \varphi \) Lipschitz, \( \Phi \) is continuous and non-decreasing. Let \( \Psi(x; \varphi) = \inf\{y : \Phi(y; \varphi) \geq x\} \); \( \Psi(\cdot; \varphi) \) is called the inverse cumulative density function, because \( \Phi(\Psi(x; \varphi); \varphi) = x \) for all \( x \in [0, L] \).
Corollary 6.10. The Earth Mover’s distance (2.9) is well defined on any road network \((R, \rho R)\).

Proof. A road network \((R, \rho R)\), being a complete and separable metric space, is therefore a Polish metric space, and also a Radon space. The Earth Mover’s distance is the same as the 1-Wasserstein distance, which is defined for all Radon spaces [6, Ch.7].

6.3 Matching Costs on Line Segments

We will use the newly constructed metric space and probability models shortly, in order to adapt the results of Chapters 3 and 5 to road networks; i.e., to vehicles which obey street constraints. A currently missing piece of that puzzle is to characterize the behavior of bipartite matchings between large random point sets on a single line segment, which can be thought of as the very simplest kind of road network. We consider such matchings in the present section.

It is argued in [43] (via “it is easy to prove”) that there are positive constants \(c\) and \(C\), \(c < C\), such that the optimal bipartite matching between two sets of \(n\) uniformly distributed points on the unit interval \([0, 1]\) has cost \(M^*(n)\) bounded by

\[
c \sqrt{n} \leq M^*(n) \leq C \sqrt{n}
\]

with high probability, i.e., with probability \(1 - o(1)\). (This order of growth can indeed be observed readily through Monte Carlo simulation.) While the present author has not found any more in-depth discussion of bipartite matching on segments whatsoever, it is easy to see that such cost should be linear in the width of the segment. In other words, for optimal bipartite matchings between uniformly distributed points on an interval of width \(W\), we may simply replace \(c\) and \(C\) by \(cW\) and \(CW\), respectively.

In this section we present as conjecture a fairly benign generalization of this law for absolutely continuous distributions with general density. The result is left as conjecture because a fully rigorous proof evaded the author; however, it is probably not difficult to prove; we examine a non-rigorous proof sketch.

Conjecture 6.11 (Matching conjecture on a segment, identical). There is a constant \(K_1\) such that for any sequence \(\{Z_k\}\) of i.i.d. random variables, with absolutely continuous distribution \(\mu\) on a real interval of length (Lebesgue measure) \(W\), the optimal bipartite matching cost \(M^*(X_1, \ldots, X_n, Y_1, \ldots, Y_n) = \min_{\sigma \in \Pi_n} \sum_{k=1}^n |X_{\sigma(k)} - Y_k|\), between its odd and even sub-sequences \(\{X_k\}\) and \(\{Y_k\}\), respectively, has limit behavior

\[
M^*(n) \leq K_1 W \sqrt{n}
\]

with high probability (i.e., with probability \(1 - o(1)\)).

Proof Sketch. Under adequate continuity assumptions for the distribution’s density function \(\varphi\), there exists a continuous and monotonic mapping \(y = g(x)\) which, when applied to \(\{Z_k\}\), obtains a sequence \(\{Z'_k\}\) of i.i.d., uniformly distributed points on the
same support. A key property we rely on is that as \( n \to +\infty \), essentially all of the matches become negligibly small. Thus, after the transformation, the average length of matches within any small interval \((y, y + dy)\) is bounded above by \( K_1 W n^{-1/2} \). (The intervals are indiscernible due to the uniform distribution, and the total matching cost is bounded by \( K_1 W \sqrt{n} \).) The total cost \( M^* \) of the un-transformed matching should have an upper bound approaching

\[
\int_x \left[ n \times \left( \frac{dy}{W} \right) \right] \frac{\text{average length of match in } (y, y + dy)}{\text{average length of match in } (x, x + dx)} \left[ \frac{dx}{dy} \right] \frac{K_1 W n^{-1/2}}{\text{number of matches in } (x, x + dx), \text{ same as in } (y, y + dy)}
\]

which, cancelling terms appropriately obtains (6.3).

What this analysis suggests is that \( \varphi \) does not really affect the constant of proportionality in front of \( \sqrt{n} \). This result is consistent with the effect of the density function in other convergence results, e.g., BHH (Theorem 2.11), specifically when \( d = 1 \).

## 6.4 Matching Costs on Roadmaps

In this section we will use a quite simple argument to relate the behavior of matchings on a line segment to the behavior of matchings on road networks. The crux of the approach is to use a closed traversal of the network (a curve that can be treated as a single line segment) as a surrogate space for measuring certain distances. Therefore, we introduce another graph problem traditionally called the Chinese Postman problem.

**Definition 6.12 (Chinese Postman problem).** The Chinese Postman problem, also known as the Route Inspection problem, is to find the minimum length closed tour traversing every edge in a graph at least once.

The tour on the road network which corresponds to the CPP tour of its roadmap will be indeed the shortest possible closed traversal of the network; it will be our surrogate.

**Proposition 6.13 (\( M^* \) on road networks, identical distributions).** If Conjecture 6.11 is true, then there is a universal positive constant \( K_1 \) such that for any road network \( \mathcal{R} \) and any sequence \( \{Z_k\} \) of i.i.d. random variables, with absolutely continuous distribution \( \mu \) on \( \mathcal{R} \), the optimal bipartite matching cost \( M^*(n) \triangleq \min_{\sigma \in \mathcal{F}_n} \sum_{k=1}^n \mathcal{P}_\mathcal{R}(Y_k, X_{\sigma(k)}) \) between its odd and even sub-sequences \( \{X_k\} \) and \( \{Y_k\} \), respectively, has limit behavior

\[
M^*(n) \leq K_1 \text{CPP}^*(\mathcal{R}) \sqrt{n} \tag{6.4}
\]

with high probability as \( n \to +\infty \) (i.e., with probability \( 1 - o(1) \)).
Proof. To obtain the proposition, let $P'$ be a cyclic tour on $\mathcal{R}$ corresponding to the optimal Chinese postman tour over the skeleton (undirected version) of its roadmap, and let $P$ be an arbitrary cut of $P'$ to remove the cycle. $P$ can be thought of as a segment of length $\text{CPP}^*(\mathcal{R})$. $P$ also covers $\mathcal{R}$, so the "visit time" function $Q: \mathcal{R} \to \mathbb{R}_{\geq 0}$ defined by $Q(p) = \inf \{ s \in \mathbb{R} : P(s) = p \}$ is well defined.

Let $\mathcal{D}_Q$ denote the metric measuring the difference between points' visit times, i.e., for any points $p, q \in \mathcal{R}$, $\mathcal{D}_Q(p, q) = |Q(q) - Q(p)|$. We can apply Conjecture 6.11 to the optimal matching according to $\mathcal{D}_Q$, since it is easy to argue that the sequence $\{Q(Z_k)\}$ is i.i.d. with an absolutely continuous distribution on $[0, \text{CPP}^*(\mathcal{R})]$. Such matching therefore satisfies (6.3) with $W = \text{CPP}^*(\mathcal{R})$. Since $\mathcal{D}_\mathcal{R} \leq \mathcal{D}_Q$ everywhere, the optimal matching according to $\mathcal{D}_\mathcal{R}$ has cost no greater, obtaining the lemma. □

Lemma 6.14 (Lower bound on $M^*$ on road networks). For any road network $\mathcal{R}$ and any sequences $\{X_k\}$ and $\{Y_k\}$ of i.i.d. random points distributed according to $f_X$ and $f_Y$, respectively, the optimal bipartite matching cost $M^*(n) := \min_{\sigma \in \Pi_n} \sum_{k=1}^n \mathcal{D}_\mathcal{R}(Y_k, X_{\sigma(k)})$ has limit behavior

$$\lim \inf_{n \to +\infty} n^{-1} M^*(n) \geq \mathcal{W}(f_Y, f_X), \quad \text{almost surely.} \quad (6.5)$$

Proof. The proof of the lemma is basically identical to the logic leading to the proof of Lemma 5.8. The only difference is that instead of using increasingly fine hyper-cube based partitions of $\mathbb{R}^d$ in the proof, one should use increasingly fine segment-wise partitions of $\mathcal{R}$. □

Unfortunately, we cannot obtain even a high-probability upper bound for the optimal matching using the techniques of Chapter 5, e.g., in the style of Lemma 5.15. The reason for this is that absolute deviations (from expectation) of the cost of the "shadow map" produced in Algorithm 2 of Chapter 3 are themselves on the order $\sqrt{n}$ of matching costs on road networks. However, we can obtain a weaker (expectation) bound:

Lemma 6.15 (Upper bound on $M^*$ on road networks). For any road network $\mathcal{R}$ and any sequences $\{X_k\}$ and $\{Y_k\}$ of i.i.d. random points distributed according to $f_X$ and $f_Y$, respectively, if Conjecture 6.11 is true, then the optimal bipartite matching cost $M^*(n) := \min_{\sigma \in \Pi_n} \sum_{k=1}^n \mathcal{D}_\mathcal{R}(Y_k, X_{\sigma(k)})$ has limit behavior in expectation

$$\lim \sup_{n \to +\infty} \frac{E M^*(n) - n \mathcal{W}(f_Y, f_X)}{\sqrt{n}} \leq K_1 \text{CPP}^*(\mathcal{R}). \quad (6.6)$$

($K_1$ is from Prop. 6.13.)

Proof. The proof of the lemma follows closely the logic behind the proof of Lemma 5.14. There are three (3) technical modifications to the proof: (i) first, we use $\sqrt{n}$ in the denominator, instead of $n^{(d-1)/d}$; (ii) second, we take an expectation before the limit; (iii) third, where Theorem 2.14 is used in that proof, we use Lemma 6.13 instead. □
6.5 Cost Bounds for SCP Tours on Roadmaps

Having derived cost bounds for bipartite matchings on road networks, we can applying those bounds using mostly the same logic as in Chapter 5 to obtain cost bounds for the SCP.

Theorem 6.16 (Lower bound on SCP* on road networks). For any road network \( R \) and any sequence \( \{(X_k, Y_k)\} \) of i.i.d. O/D pairs, with distribution \( f \) over \( R^2 \), the optimal stacker crane tour through the first \( n \) pairs has limit behavior

\[
\liminf_{n \to +\infty} \frac{1}{n} \text{SCP}^*(n; f) \geq \mathcal{M}(f), \quad \text{almost surely.} \quad (6.7)
\]

Proof. Follows the proof of Theorem 3.2 in Chapter 5. \( \square \)

Again, the strongest upper bound we can obtain by the techniques of Chapter 5, this time for the SCP, is an expectation bound.

Proposition 6.17 (Upper bound on SPLICE on road networks). Let \( A \) be any SPLICE algorithm with subtours connected together by fragments of a CPP tour. If Conjecture 6.11 is true, then for any road network \( R \) and any sequence of i.i.d. O/D pairs \( \{(X_k, Y_k)\} \), with distribution \( f \) over \( R^2 \), the optimal stacker crane tour through the first \( n \) pairs has limit behavior

\[
\limsup_{n \to +\infty} \frac{\mathbb{E}[A(n; f)] - n \mathcal{M}(f)}{\sqrt{n}} \leq 2K_1 \text{CPP}^*(R). \quad (6.8)
\]

(\( K_1 \) is from Prop. 6.13.)

Proof. Let \( \mathcal{X}_n \) and \( \mathcal{Y}_n \) denote the origin and destination sets, respectively. One can write

\[
A(n) \leq M^1(\mathcal{X}_n, \mathcal{Y}_n) + M^*(\mathcal{X}_n, \mathcal{X}_n) + \text{CPP}^*(R)
\]

\[
= \left( M^1(\mathcal{X}_n, \mathcal{Y}_n) - n \mathbb{E}_{J} \mathcal{D}_R(X, Y) \right)
\]

\[
+ \left( M^*(\mathcal{Y}_n, \mathcal{X}_n) - n \mathcal{M}(f_Y, f_X) \right)
\]

\[
+ n \mathcal{M}(f) + \text{CPP}^*(R). \quad (6.9)
\]

The claim is obtained—as for Theorem 5.17—by moving the second-to-last term (Mover’s complexity) to the left-hand side of the inequality, dividing on both sides by \( \sqrt{n} \), and taking the limit as \( n \to +\infty \). (Before the limit, though, we take the expectation.) Only the optimal matching term does not vanish, and applying Lemma 6.15 obtains the proposition. (The extra factor of 2 in (6.8) is because of Remark 5.16.) \( \square \)

Remark 6.18. It is clear that Proposition 6.17 holds for the optimal SCP tour and for any SPLICE + c-TSP algorithm as well. Those algorithms produce stacker crane tours no longer than those of the SPLICE + CPP algorithm, surely.
6.6 Dynamic Taxiing with Generally Distributed Random Demands on Roadmaps

Finally, having derived cost bounds like those in Chapter 5, for large SCP tours through random demands, we can obtain extensions of the results from Chapter 3 about the dynamic case, i.e., the 1-DPDP. Fortunately, the expectation bound of the previous section is sufficient to derive stability conditions and performance bounds by the same basic logic.

We will define the Stochastic Queue Medians and Stacker Crane policies, in the road network setting, using exactly the same descriptions appearing in Sections 3.5.1 and 3.5.2, respectively. Those descriptions are geometry-agnostic, although we will soon demonstrate that the growth rate of the system time bounds is not.

It is straightforward to show that the SQM policy system time bound of Theorem 3.6, and also the policy-independent lower bound of Theorem 3.13, hold without modification in light traffic load conditions. Therefore, we need only re-consider the heavy-load case.

6.6.1 Analysis of the Stacker Crane Policy in Heavy-load

The performance of the Stacker Crane policy in heavy load is characterized by the following theorem.

**Theorem 6.19 (Stacker Crane policy on road networks, heavy load).** There is a universal constant $K_1$ (if Conj. 6.11 is true) such that the system time $T_{SC}$ under the Stacker Crane policy on a road network $\mathcal{R}$ satisfies

$$\lim_{\rho \to 1} \frac{T_{SC} (1 - \rho)^2}{4K_1^2 \text{CPP}^*(\mathcal{R})^2} \leq \frac{\text{CPP}^*(\mathcal{R})^2}{mvM(f)}. \quad (6.10)$$

**Proof.** The proof of the theorem is by combining (6.8), as a bound for the optimal SCP tour, with (3.19), and applying Lemma 3.9 (with $\gamma = 1/2$).

It is noteworthy that the bound for the Euclidean planar case (e.g., Theorem 3.7 with $d = 2$) does not differ drastically in its rate of growth from the bound for road networks.

6.6.2 Lower Bounds

**Theorem 6.20 (System time, heavy load bound).** Within the class of origin-fair policies, for the 1-DPDP on any road network $\mathcal{R}$ with a demand distribution such that $\varphi_Y = \varphi_X = \varphi$,

$$\lim_{\rho \to 1} \frac{T^* (1 - \rho)}{4mv} = \frac{|\mathcal{R}|}{4mv}. \quad (6.11)$$

**Proof.** Since road networks are almost everywhere locally like $\mathbb{R}^1$ (i.e., except at the intersections), Lemma 3.16 holds as is for road networks. (The neighborhood-based
arguments of the proof of Lemma 3.16 are inviolate for the conditioning on all points except the intersections; however, under our assumption of absolutely continuous distributions, the intersection points have zero measure.) Therefore, following the same logic as the proof of Theorem 3.15, we obtain (3.23) for unbiased policies on road networks, with \( d = 1, V_1 = 2, \) and \( \gamma_1 = 1/4 \) (\( \gamma_d = \frac{d}{d+1} V_d^{-1/4} \)).

Of course, this lower bound holds again in the case that \( f_y = f_x \) only. The bound is also not tight with respect to the upper bound, and it remains an open question (for the same reasons as in the planar case) to obtain a lower bound and upper bound that match.

**Proposition 6.21.** For causal and stationary, but potentially spatially unfair policies, the system time lower bound is

\[
\lim_{\theta \to 1-} \bar{T}^*(1 - \theta) \geq \frac{(\int_{\mathcal{R}} \varphi^{1/2}(x) \, dx)^2}{4mn}.
\]

**Proof.** For the same reasons, we obtain the result by substituting \( d = 1 \) and \( \gamma_1 = 1/4 \) into (3.29) of Prop. 3.17.

### 6.6.3 Stability Conditions

**Theorem 6.22 (Stability condition on road networks).** Consider the 1-DPDP defined in Section 3.2, set in a road network environment \( \mathcal{R} \). The condition \( \theta < 1 \) is necessary and sufficient for the existence of stabilizing policies.

**Proof.** The proof that \( \theta < 1 \) is necessary is by the same logic as in the proof of Theorem 3.11, except that the length of the stacker crane tour is governed by Theorem 6.17 on road networks. Using the new expression does not change the result. The proof that \( \theta < 1 \) is sufficient is again provided by the analysis of the Stacker Crane policy, which, according to Theorem 6.19, will stabilize the theater as long as \( \theta < 1 \).

### 6.7 Simulation Study

#### 6.7.1 Performance of the Stacker Crane policy

In this section, we present simulation results with evidence to support the theoretical assertions of the chapter: mainly, that of Theorem 6.19. Thus, again, we focus mainly on the asymptotics of the heavy-load scenario.

The plot in Figure 6-6 shows the (empirical) average system time versus system utilization \( \theta \), for 1-DPDP simulations with a single, unit-speed vehicle. In these simulations, origins and destinations alike were sampled i.i.d., uniformly distributed over the “square” map of Figure 6-5; each road (NESW) has unit length. As in the plots of Section 3.9.1, each point represents a single 1-DPDP simulation, run for sufficiently long that the per-demand average system time “stabilized”. The x-axis
Figure 6-5: A square road network with roads: North (N), East (E), South (S), and West (W), all unit length.

of the plot is derived from the left-hand side of (6.10), with the goal of obtaining [convergence to] a linear trend. The top axis is annotated with the corresponding utilization values, in a range of \(0.87-0.95\). (Note that these values are considerably higher than those simulated in the Euclidean setting of Chapter 3.) In this regime, the average system time indeed is quite well predicted by a linear trend, as shown by a dashed fit line.

\[ \phi = \frac{\lambda M}{mv} \]

Figure 6-6: Average system time versus workload under the Stacker Crane policy, with \(m = 1\), unit-speed vehicle on the square roadmap. Origins and destinations all i.i.d., uniformly distributed.

Figure 6-7(a) shows a plot of the average system time under the SC policy, with fleet size \(m\) in the range 1–10, for a fixed utilization factor \(\phi = 0.942\). Again, by fixing \(\phi\), we are letting the bound of Theorem 6.19 scale as \(1/m\). Figure 6-7(b) shows the same data on a log-log scale. Although the average system time does not reduce by the full decade (predicted) from \(m = 1\) and \(m = 10\), it is close, e.g., as compared to the trend of Figure 3-5(b). We may suspect that as the fixed workload \(\phi\) approaches 1−, the slope of this trend may approach the predicted one of −1.
6.7.2 Stability Conditions

In this section we present again results of the simulation study first described in Section 3.9.2, to measure the critical threshold $\lambda^*$, separating stabilizable arrival rates from unstabilizable ones (given a fixed setting of the other system parameters). Of course, now our focus is on the homogeneous taxi theater on a roadmap. As in Section 3.9.2, the results are meant to validate our theoretical findings and demonstrate the role of the Earth Mover’s distance in predicting maximum throughput rates for vehicle sharing systems modelled by the 1-DPDP on roadmaps.

Our simulations were of the Stacker Crane policy, which we have shown to be perfectly stabilizing. In each iteration, the stacker crane tour was computed using the LARGEARCS algorithm, with a new bipartite matching algorithm which we will present in Chapter 7; the entire algorithm was implemented in Python. (Simulations were run on an Intel i5 processor with 4 CPUs and 4GB of RAM.)

A simulation experiment was repeated for fifty (50) randomly-generated scenarios, each characterized by (i) a randomly generated, connected roadmap $R$ of 1–10 roads, (ii) a randomly generated demand distribution (with randomized but constant density per pair of roads), and (iii) a randomly sized fleet of between 1–5 unit speed vehicles. The critical rate $\lambda^*$ was computed by $m/.\mathcal{M}(f)$. The Earth Mover’s distance component $\mathcal{W}(f_Y, f_X)$ was computed using a new technique which we will introduce in Chapter 8. The expected carry time $\tau = \mathbb{E}_f \mathcal{D}_R(X, Y)$ was computed using an explicit formulation which is the subject of a forthcoming paper (Monte Carlo averaging is another viable option to estimate this statistic). In each case, the arrival rate simulated was $2\lambda^*$ (exceeding theoretical capacity by 100%), and the simulation was run for $T = 1000$ time. Figure 6-8 shows a very strong corroboration between the predicated and empirical per-demand average service times $\bar{s}$.

In addition to the randomized scenarios, we considered also a particular toy distribution. We let the demand distribution $f$ be defined as follows: (i) With probability
Figure 6-8: Empirical distribution of average service times (observed vs. predicted) over many random pairs of roadmap and demand distribution.

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>N</td>
<td>1/5</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>1/5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2/5</td>
</tr>
</tbody>
</table>

Table 6.3: Probability mass function (pmf) $\mu(r_1, r_2)$, associated with “toy” demand distribution.

given by Table 6.3, $X \in r_1$ and $Y \in r_2$ (on the roadmap shown in Figure 6-5); (ii) given their road assignments, the coordinates of $X$ and $Y$ are independent and uniformly distributed over $[0, 1]$. Under such distribution, the expected carry distance is equal to $17/15$, the Earth Mover’s distance is equal to $31/30$, and the Mover’s complexity (the sum of the two terms) is the predicted average per-demand service time $\bar{s} = 13/6 \approx 2.167$.

Figure 6-9 shows plots of the number of outstanding demands over the duration $T = 10,000$ of two experiments with different arrival rates: Figure 6-9(a) shows the result of the experiment with arrival rate $\lambda = 0.99 \times \lambda^*$, which is below the stabilizable threshold. The resulting plot includes several “renewals” (times when the system is empty) and does not exhibit uncontrolled growth in the number of outstanding demands. Figure 6-9(b) shows the result of the experiment with arrival rate $\lambda = \lambda^* + 0.1$. The number of outstanding demands reaches $\approx 1,000 = 0.1 \times T$ by the final time, showing strong corroboration of our predictions.

6.8 Conclusion

In this chapter, we began Part II of the thesis, which emphasizes a more practical study of taxi-like transportation systems, i.e., shared-vehicle, one-way service, with limited carrying capacity. Our main approach to the practical study has been to re-consider the fundamental results developed in Chapters 3 and 5, but with a rich and arguably more appropriate framework for modeling the street constraints usually
imposed on vehicles for physical transport. Therefore, we have defined formally a class of one-dimensional metric spaces which are $\mathbb{R}^1$-like but may have arbitrary, graph-like topology.

In Sections 6.4, 6.5, and 6.6, we re-derived versions of all the main results from Chapters 3 and 5, under the new environment models. The forms of these new results are quite reminiscent of those derived in the Euclidean case, e.g., the EMD and the Mover’s complexity carry out the same role.

One limitation of the study is an implicit assumption that vehicles can travel forward or backward on a particular road segment at the same maximum speed, and that they may change direction at will, instantaneously. Placing additional restrictions to restrict this type of motion is quite likely to have a non-negligible effect on travel times.

Figure 6-9: Trajectories of number of outstanding demands over time ($T = 10000$), in stable and unstable regimes (6-9(a) and 6-9(b), respectively).
Chapter 7

Fast Bipartite Matching with Roadmap Distances

7.1 Introduction

Let us recall the Bipartite Matching problem, or Assignment problem [30]. Given two sets \( S = (s_1, s_2, \ldots, s_M) \) and \( T = (t_1, t_2, \ldots, t_M) \), of \( M \) points each from a domain \( \Omega \), a matching is a set of pairs \( \mathcal{A} \subseteq S \times T \) such that each point appears in \( \mathcal{A} \) exactly once. Every matching represents a unique bijective mapping of \( S \) onto \( T \), and therefore, is uniquely determined by some permutation \( \sigma \) of \( (1, 2, \ldots, M) \), in the sense that \( \mathcal{A} = \{(s_k, t_{\sigma_k})\}_{k=1}^M \). Given a distance metric \( \mathcal{D} : \Omega^2 \to \mathbb{R}_{\geq 0} \), we define the cost of a match \( (s, t) \in \mathcal{A} \) as \( \mathcal{D}(s, t) \) and the cost of the matching \( \mathcal{A} \) as the sum over pairs \( \sum_{(s, t) \in \mathcal{A}} \mathcal{D}(s, t) \). The Assignment problem is to find the minimum cost, or minimal, matching.

As discussed in Chapter 5, computing bipartite matchings dominates the runtime complexity of the LARGEARCS algorithm on points in Euclidean \( \mathbb{R}^d \) (\( d \geq 2 \)), part of the best known constant-factor algorithm [57] for the Stacker Crane problem (SCP). The SCP is a NP-Hard problem whose objective is to obtain the smallest tour through many one-way transportation demands. We showed in Chapter 5 that LARGEARCS is asymptotically optimal for large randomly generated instances (almost surely). Therefore, efficient algorithms for matching are desirable for dealing with large SCP instances.

Literature Review: While assignment problems have numerous uses for solving and approximating other combinatorial optimization problems, they also have more direct applications in domains such as operations management, computer science, computational biology, and computational music. For example, they can be used to represent the problem of scheduling work to assets optimally (e.g., programs to computer processors, or jobs to machines or workers). In computational biology, the many-to-one assignment problem on a line is used to solve the Restriction Scaffold Assignment problem [38], an important step which is solved millions of times in DNA sequencing. In computational music, assignment costs are used to measure the similarity of musical rhythms [44].
In general, the assignment problems can be solved in $O(M^3)$ time using the classical Hungarian method [73]. If the points are in $\mathbb{R}^2$ and the distance between them is Euclidean, then there is a class of algorithms [3] for the one-to-one assignment, or bipartite matching problem, which achieve $O(M^{2+\epsilon})$ time for any $\epsilon > 0$. If the points are on a line, then there is a trivial $O(M \log M)$ algorithm to solve the one-to-one assignment problem, the optimality of which is proved, e.g., in [127]. For the case that the points lie on a circle, another $O(M \log M)$ algorithm was given originally by Karp and Li [70], which has been refined somewhat by others, e.g., [128]. Colannino et. al. provided companion $O(M \log M)$ algorithms on the line for both the many-to-one [38] and many-to-many [39] versions of the assignment problem; both algorithms are fundamentally based on the original insights developed by Karp about the circle, though they provide non-trivial additional insights themselves. (The algorithms on the line are actually $O(M)$ if the points are already sorted; however, on the circle the runtime remains $O(M \log M)$.) The best known many-to-one and many-to-many algorithms on the circle remain $O(M^2)$.

Contribution: In this chapter, we provide an algorithm which computes the minimal matching between two sets of $M$ points each, on a fixed but arbitrary road network (as defined in Chapter 6) with $m$ roads and $n$ vertices. The runtime of the algorithm is $O(M \log M) + \log M \times O(m(m + n \log n))$, which for a fixed roadmap is dominated asymptotically by the term $O(M \log M)$. It is the best runtime bound achievable in terms of $M$, and it significantly improves on the $O(M^3)$ general algorithm. The design of the algorithm is influenced by the elegant treatment of bipartite matching on lines and circles in [128], which are the two roadmap structures one can construct from a single road ($m = 1$). The crucial component of the algorithm is the application of a capacity-scaling approach for the minimum convex cost flow problem, developed, e.g., in [4, Sec. 14.5].

Organization: The rest of the chapter is organized as follows: In Section 7.2 we review the existing results for bipartite matching on line segments and circles, which are the two most basic roadmaps. We state the problem of finding the minimal bipartite matching between points on a general roadmap rigorously in Section 7.3. In Section 7.4 we generalize the analysis of bipartite matching on line and circles and introduce an optimal algorithm for bipartite matching on general roadmaps. We discuss computational complexity in Section 7.4.4 and demonstrate that our algorithm can be computed in $O(M \log M)$ time. We provide some discussion of the results and closing remarks in Section 7.5.

7.2 Optimal Matching on Lines and Circles

If the domain $\Omega$ is a single line segment with distance defined by $\mathcal{D}(s, t) = |t - s|$, and both $S$ and $T$ are sorted, then it is well known (shown, e.g., in [127]) that the minimal matching is determined by the identity permutation $\sigma$, i.e., $\mathcal{A} = \{ (s_1, t_1), (s_2, t_2), \ldots, (s_M, t_M) \}$. Therefore, if $S$ and $T$ are already sorted, the minimal matching can be transcribed in $O(M)$ time. If the points are not sorted, then sorting them first takes $\Theta(M \log M)$ time in the worst case.
Such a straightforward scenario would seem to warrant little additional discussion, but [128] presents a characterization of the cost of the minimal matching, which becomes useful in various extensions: Let $[0, L)$ be the interval containing $S$ and $T$, and let us orient all matches $(s, t)$ from $s$ to $t$. If $s < t$ we say the match is forward, because the orientation of the match is in the direction of the positive coordinate axis; if $s > t$, then we say the match is backward. Let $H(y)$ denote the total number of forward matches crossing a coordinate $y$ minus the total number of backward matches crossing $y$. A matching $A$ is called unidirectional if at every such coordinate, all of the matches crossing the point have the same orientation.

**Lemma 7.1.** Every minimal matching is unidirectional.

*Proof.* A simple proof is given in Lemma 3 of [128], but we provide another proof that generalizes readily to every scenario in this chapter. The proof is by contradiction: Assume that $A^*$ is a minimal match but there is some coordinate $y$ crossed by $M^+ > 0$ forward matches and $M^- > 0$ backward matches. Consider the case that there are no points at $y$. If $y$ has points, then we simply ignore all the matches that start or end there (they do not cross). With similar justification, assume that $y$ is an interior point. Then there is a neighborhood $(y - \epsilon, y + \epsilon)$ for some $\epsilon > 0$ which contains no points and is crossed by exactly the matches through $y$. Choose one of the forward matches and one of the backward matches and exchange their endpoints in $T$. The resulting matching has cost at least $4\epsilon$ less than the minimal matching $A^*$.  

Note by Lemma 7.1 that the total number of matches crossing a point $y$ is $|H(y)|$ for the minimal matching $A^*$; if $H(y) > 0$ then they are forward matches; otherwise, they are backward matches. Thus, the total length of all matches, i.e., the cost of the optimal matching, can be written as

$$\int_0^L |H(y)| \, dy. \quad (7.1)$$

Let $N_S(y)$ denote the number of points $s \in S$ such that $s < y$, i.e., $|S \cap [0, y)|$, let $N_T(y)$ denote the number of points $t \in T$ such that $t < y$, i.e., $|T \cap [0, y])$, and let $F(y) = N_S(y) - N_T(y)$. Note that since $|S| = |T|$ we have $F(0) = F(L) = 0$. As argued in [128], $F(y) \equiv H(y)$ on a line segment, because whenever we cross a point $s \in S$ in the positive direction, either a forward match is beginning, or a backward match is ending; either way, both $H$ and $F$ increase by 1. (Whenever we encounter a point $t \in T$, both $H$ and $F$ decrease by 1.) Therefore, the cost of the minimal matching can be computed without producing one, by computing $F$ and substituting it in (7.1).

Now suppose that $\Omega$ is a circle instead of a line; for example, the circle of unit radius $[0, 2\pi)$ with distance between points $y_1$ and $y_2$ defined by $\min\{|y_1 - y_2|, 2\pi - |y_1 - y_2|\}$. Though the circular case is quite similar to the linear one, the identity permutation is not necessarily minimal for sorted $S$ and $T$, because matches across the $y = 0$ boundary are now possible. It is easy to argue, however, that the minimal matching is among the $M$ circular shift permutations, which leads immediately to an
\(O(M^2)\) minimal matching algorithm, e.g., the one in [127]. The \(O(M^2)\) barrier was indeed broken in [73], where the authors observed that \(H(y) = F(y) + z\) for some integer \(z\). Since \(F(0) = 0\), then \(z = H(0)\), which means \(z\) can be thought of as the (signed) number of matches crossing \(y = 0\). Now the minimal matching has cost

\[
C(z) = \int_0^{2\pi} |F(y) + z| \, dy.
\] (7.2)

Certainly, the cost of the optimal matching can be no less than the minimum value of (7.2) taken over integer \(z\). [128] provides an \(O(M \log M)\) algorithm to compute such minimum and an \(O(M)\) follow-up procedure to produce a matching of cost no greater than \(C(z)\).

### 7.3 Problem Statement

The objective of the chapter is to obtain an algorithm for the bipartite matching problem on roadmaps which for a fixed roadmap \(R\) has worst-case runtime bounded by \(O(M \log M)\), where \(M\) is the number of points in each of \(S\) and \(T\).

**Lemma 7.2 (Lemma 2 of [128]).** \(\Omega(M \log M)\) is a lower bound for the time it takes to find the minimal cost matching in both the linear and the circular cases.

**Proof Sketch.** The lemma was proved by a reduction of the Set Equality problem, as in [128]: Suppose the cost of a match between two elements is 0 if the elements are equal, and strictly positive otherwise; e.g., one could map every distinct element to a unique position on the line segment \([0, 1]\) and take the cost of a match as the distance between the points. Then two sets are equal if they admit a zero-cost bipartite matching. The Set Equality problem is known to be \(\Omega(M \log M)\).

\(\Omega(M \log M)\) is clearly a lower bound for the time to find a minimal cost matching on a fixed but arbitrary roadmap, because lines and circles are only two specific kinds of roadmaps.

### 7.4 Optimal Bipartite Matching on a Roadmap

In this section, we generalize the analysis of [128], about the cost of matchings on lines and circles, for the case of any fixed but arbitrarily complex roadmap \(R\). We assume that the cost of a match is equal to the roadmap distance between the points, i.e., the length of the shortest path between them. The result of our analysis will provide insight about the design of a novel minimal matching algorithm.

#### 7.4.1 Cost Characterization

Since shortest paths are minimal, they are also simple (they do not "cross themselves"). We will attribute to every match \((s, t)\) the orientation of the shortest path...
from \( s \) to \( t \) on \( \mathcal{R} \). Note that a match need not have the same orientation on every road on such a path; wherever its path follows a road \( r \) in the positive direction, we say it is forward; wherever it follows a road in the opposite direction, we say it is backward. Without loss of generality, we assume that shortest paths are unique, e.g., by perturbation or arbitrary tie-breaking.

We will now re-define the quantities \( F \) and \( H \) for roadmaps: For every road \( r \in R \), \( 0 \leq y \leq L_r \), let \( N_S(y; r) \) denote the number of points \((r, s_i)\) in \( S \) which are on \( r \) such that \( s_i < y \), and let \( N_T(y; r) \) denote the number of points \((r, t_i)\) in \( T \) which are on \( r \) such that \( t_i < y \). Let \( F(y; r) = N_S(y; r) - N_T(y; r) \). Note that \( F(0; r) = 0 \) for all \( r \in R \), but now we have \( F(L_r; r) = |S \cap r| - |T \cap r| =: b_r \neq 0 \) in general. Let \( b_r \) be called the surplus of road \( r \).

Given a unidirectional matching \( A \), let \( H(y; r) \) denote the number of forward matches crossing the coordinate \((r, y)\), minus the number of backward matches crossing it. Using the same arguments in [128], it is easy to argue that \( H(y; r) = F(y; r) + z_r \) for some integer \( z_r \); there is one such integer for every road \( r \in R \), and so we can write the total cost of all the match fragments using road \( r \) as

\[
C(z_r; r) = \int_0^{L_r} |F(y; r) + z_r| \, dy. \tag{7.3}
\]

It will be convenient in the sequel to relax the discrete nature of (7.3) with respect to \( z_r \), i.e., let \( z_r \) be real-valued. The resulting collection of continuous functions (one for each \( r \in R \)) share several convenient properties which will be crucial to the analysis and algorithm design in the next sections.

**Lemma 7.3 (Cost properties).** Letting \( C \) be defined as in (7.3) for any \( r \in R \) and all \( z_r \in \mathbb{R} \) (and not just the integers), \( C \) is (i) piece-wise linear, (ii) convex, and (iii) unbounded, and (iv) has constant slope in every integer interval.

**Proof.** We prove the result for a single road \( r \), omitting the \( r \)-specific notation. Note that \( F \) is piece-wise constant and takes value only on the set of integers. Let \( F = \{f_1, \ldots, f_m\} \subset \mathbb{Z} \) denote the set of values taken by \( F \) (in increasing order),
and let $I_1, \ldots, I_{m+1}$ denote the intervals $(-\infty, f_1), (f_1, f_2), \ldots, (f_{m-1}, f_m), (f_m, +\infty)$, disjunctly covering $\mathbb{R}$. $C(z)$ can be written as

$$C(z) = \int_{F(y)+z>0} [F(y) + z] \, dy - \int_{F(y)+z<0} [F(y) + z] \, dy.$$  \hfill (7.4)

For each $k = 1, \ldots, m+1$ and all $z \in -I_k$ (i.e., $-z \in I_k$), we have that $\{y : F(y) + z > 0\} \equiv \{F(y) > f_{k-1}\} =: \mathcal{Y}_k^+$ and $\{y : F(y) + z < 0\} \equiv \{F(y) \leq f_{k-1}\} =: \mathcal{Y}_k^-$. Define scalar constants

$$\alpha_k \doteq \int_{\mathcal{Y}_k^+} F(y) \, dy - \int_{\mathcal{Y}_k^-} F(y) \, dy,$$

and

$$\beta_k \doteq |\mathcal{Y}_k^+| - |\mathcal{Y}_k^-|,$$  \hfill (7.5)

then for all $z \in -I_k$, (7.4) can be written as $C(z) = \alpha_k + z\beta_k$.

The prequel demonstrates that $C$ is piece-wise linear with constant slope within every integer interval. It is obvious that $C$ is continuous. Thus, to show that $C$ is convex, one can show that the slope of $C$ is non-decreasing, e.g., by confirming that $\beta_k$ is non-increasing in $k$. Unboundedness can be proved by confirming that $\beta_1 = L$ and $\beta_{m+1} = -L$, where $L > 0$ denotes the length of the road. \hfill $\square$

### 7.4.2 Cost Bounds on Optimal Matchings

The next problem can be used to bound the cost of the minimal matching from below, and we will eventually demonstrate that its solution can be used to produce a matching which achieves the bound.

**Problem 7.1 (Minimum cost of matching).**

$$\begin{align*}
\text{minimize} & \quad C(Z) = \sum_{r \in R} C(z_r; r) \\
\text{subject to} & \quad \sum_{\{r \mid r^+ = u\}} z_r + b_u = \sum_{\{r \mid r^- = u\}} z_r \quad \text{for all vertices } u.
\end{align*}$$  \hfill (7.6) \hfill (7.7)

**Lemma 7.4.** The minimal matching has cost bounded below by (7.6) of the optimal solution of Problem 7.1.

**Proof.** Given any unidirectional matching $A$, one can compute $z_r := H(\cdot; r) - F(\cdot; r)$ for each $r \in R$. The cost of $A$ is precisely (7.6) with the vector $Z \in \mathbb{Z}^R$ composed of such $z_r$. For each $u \in \mathcal{V}$, the left hand side of (7.7) is equal to the number of matches entering $u$, and the right hand side is equal to the number of matches leaving $u$ (both signed counts). Regular conservation arguments imply that (7.7) must hold for any feasible matching. Therefore, the feasible set of Problem 7.1 contains all $Z$ realizable by unidirectional matchings, which contain the minimal matchings, obtaining the lemma. \hfill $\square$
The previous formulation can be extended quite easily to incorporate the notion of “one-way” roads: For example, if a road \( r \in R \) admits only forward matches, that condition can be encoded as \( H(y; r) \geq 0 \) for all \( y \in (0, L_r) \), i.e., \( \min_y F(y; r) + z_r \geq 0 \). Note that the framework can support one-way and bi-directional roads simultaneously.

**Lemma 7.5 (Integral solutions).** The optimization problem Problem 7.1 has integer optimal solutions.

The significance of Lemma 7.5 is that we may hope to avoid the complexity associated with integer optimization problems by linear relaxation.

**Proof.** Suppose that \( Z \) is a fractional optimal solution to (7.1), and let \( r_1 \) be a road with non-integer component, i.e., \( z_{r_1} \notin \mathbb{Z} \). Note that since all the coefficients of (7.7) are integer, then among the roads which share endpoint \( r_1^+ \) with \( r_1 \) (including itself if \( r^- = T' \)), at least one must also be fractional. Using this argument, one can identify a cycle \((r_1, \ldots, r_K)\) of only roads with fractional components. For each \( k = 1, \ldots, K \), let us denote by \( I_k \subset \mathbb{R} \) the unit interval \([z_{r_k}, z_{r_k}]\), and let \( \Pi \) denote the unit hyper-cube \( \prod_k I_k \). Note that \((z_{r_1}, \ldots, z_{r_K})\) is in the interior of \( \Pi \). In the interior of \( \Pi \), we have a gradient

\[
\nabla_{(z_{r_1}, \ldots, z_{r_K})} C(Z) = \left[ \frac{d}{dz} C(z; r_1) \bigg|_{z_{r_1}} \ldots \frac{d}{dz} C(z; r_K) \bigg|_{z_{r_K}} \right].
\]

Applying Lemma 7.3, it is clear that such gradient is constant over \( \Pi \). In particular, the gradient must be 0, since \( Z \) is optimal by assumption. Let \( g \in \mathbb{R}^R \) denote the vector which is +1 for each road traversed in the positive direction by the cycle, \( -1 \) for each road traversed in the reverse direct, and 0 for all roads not in the cycle. (It is easy to show that \( Z + \alpha g \) satisfies (7.7) for all \( \alpha \in \mathbb{R} \).) One can find \( \alpha \in \mathbb{R} \) such that \( Z + \alpha g =: Z^+ \) lies on the boundary of \( \Pi \). \( Z^+ \) is also feasible optimal, and has at least one less fractional component than \( Z \). Such procedure can be repeated until an integral optimal solution is obtained. \( \square \)

**Solving Problem 7.1**

Since by Lemma 7.3 the objective functions \( C(\cdot; r) \) are all convex, Problem 7.1 is a so-called [minimum] convex cost flow problem [4, Ch. 14]. The convex cost flow problem (CCFP) is a generalization of the minimum [linear] cost flow problem, such that the edge costs needn’t be linear. Ahuja et. al. provide a capacity-scaling algorithm for CCFPs like Problem 7.1 which have integral solutions [4, Sec. 14.5]. Provided that the objective functions can be evaluated in \( O(1) \) time, the algorithm obtains runtime \( O((m \log U)S(n, m, C)) \), where \( n \) and \( m \) are the number of vertices and edges in the network, respectively, \( U \) and \( C \) are bounds on the total supply and cost, respectively, and \( S(n, m, C) \) is the time to solve the shortest path problem on such a network [4, Ch. 4]. Choosing, for example, the strongly polynomial \( O(m+n \log n) \) Fibonacci heap shortest-path algorithm of Fredman and Tarjan [59], and observing that \( U = M \) is an acceptable supply bound, one can solve Problem 7.1 in \( O(m \log M(m + n \log n)) \) time. Moreover, the algorithm always provides integer solutions.
Obtaining an Optimal Roadmap Matching

In this section, we provide an algorithm for the construction of an optimal matching given the vector $Z^*$. The algorithm generalizes, in a fairly clean way, e.g., the procedure given in [128].

Given a solution $Z$ to Problem 7.1, the following procedure will obtain a matching with cost less than or equal to $C(Z)$: The procedure is in two steps. During the first step, one obtains an intermediate data structure, of an integer-weighted digraph $G(Z)$ on vertex set $S \cup T \cup V$. In such graph, every edge corresponds to an empty interval on the roadmap (generally, one spanning the space between two adjacent points); thus, we will call $G(Z)$ the interval graph. In the second step, to obtain the matching itself, we run a simple graph traversal algorithm (Algorithm 4) on $G(Z)$.

The edges of the interval graph are directed, and each is labeled with a positive integer weight, indicating the number of matches which cross the corresponding interval. For example, let $(y_1, y_2, \ldots, y_K)$ be the ordering of the coordinates of points in $S \cup T$ which are on some road $r$. Let $Y_1, Y_2, \ldots, Y_{K+1}$ denote the set of intervals $(0, y_1), (y_1, y_2), \ldots, (y_{K-1}, y_K), (y_K, L_r)$, and let $f_k$ denote the constant value taken by $F(\cdot; r)$ per interval $Y_k$. Note that $f_k + z_r =: h_k$ is the signed number of matches traversing $Y_k$ under $Z$. If $h_k > 0$ then $G(Z)$ contains the edge from the earlier endpoint $(r, y_{k-1})$ to the later endpoint $(r, y_k)$; if $h_k < 0$, then it has the reverse edge; if $h_k = 0$, then there is no edge. If the edge is contained, then it has weight $|h_k|$ and length equal to the interval length. Note that the endpoints $(r, 0)$ and $(r, L_r)$, of intervals $Y_{1}$ and $Y_{K}$, respectively, are not points in $S \cup T$. They are substituted in $G(Z)$ by the roadmap vertices $r^- \in V$ and $r^+ \in V$, respectively, to which they correspond.

Remark 7.6. Note that $C(Z)$ is equal to the sum over all edges in $G(Z)$ of weight $\times$ length.

Algorithm 4 produces a matching by visiting every vertex in $G(Z)$, among which are all of $S \cup T$. When the current vertex $i$ is a point in $S$, then it is “collected” for future use. When $i \in T$, then it is matched to a point $j$ previously collected. To ensure that there are always enough points collected for future matches, $G(Z)$ is traversed in a topological order.

Definition 7.7 (Topological ordering). A topological ordering of a directed acyclic graph $G = (V, A)$ is a linear ordering $\leq$ of the vertex set $V$, so that for every $a \in A$ it holds $a^- \leq a^+$.

It is know that any directed acyclic graph (DAG) has at least one topological ordering. An algorithm to compute a topological ordering of a DAG in time $O(|V| + |A|)$ can be found in [40, Sec. 22.4].

Lemma 7.8. The cost of the matching produced by Algorithm 4 on interval graph $G(Z)$ has cost no greater than $C(Z)$.

Proof. During the execution of Algorithm 4, every point $s \in S$ traverses a path in $G(Z)$ to its match $A(s) \in T$. The cost of the matching produced cannot be greater
Algorithm 4 ConstructMatching

Input: an interval di-graph $G$ (e.g., generated from inputs $R$, $S$, $T$, and $Z$)
Output: a bipartite matching $A$ between $S$ and $T$

1: initialize: $A \leftarrow$ an empty matching
2: initialize: Associate with each vertex $v \in S \cup T \cup V$ an empty set $L_v$.
3: Choose any topological ordering of $S \cup T \cup V$ under $G(Z)$. Enumerate $S \cup T \cup V$ in this order:
4: for all vertices $v$ do
5: If $v \in S$, then add $v$ to $L_v$. Otherwise, if $v \in T$ then remove some point $u$ from $L_v$, and insert the match $(u, v)$ into $A$. If $v \in V$, we do not alter $L_v$, nor create a match.
6: For every edge $(v, w)$ leaving $v$, move $weight(v, w)$ elements from $L_v$ into $L_w$.
7: end for
8: return $A$

Algorithm 4 has one technical caveat: It assumes that a topological ordering exists under $G(Z)$. This is guaranteed as long as $G(Z)$ is a DAG, since every DAG has at least one topological ordering.

Lemma 7.9. Let $Z^*$ be an optimal solution to Problem 7.1. $G(Z^*)$ is a DAG.

Proof. The proof is by contradiction. Suppose $Z^*$ is optimal, but $G(Z^*)$ has a directed cycle $\mathcal{C} = (a_1, \ldots, a_K)$. $\mathcal{C}$ corresponds to a cycle $\mathcal{C}' = (r_1, \ldots, r_K)$ of roads in $R$. Let $g \in \mathbb{R}^R$ denote the vector which is +1 for each road traversed in the positive direction by $\mathcal{C}'$, -1 for each road traversed in the reverse direction, and 0 for all roads not in $\mathcal{C}'$. $Z^* - g$ is feasible because it satisfies (7.7). It is easy to argue that we can obtain $G(Z^* - g)$ from $G(Z^*)$ by subtracting 1 from the weight on every edge in $\mathcal{C}$. (We remove any edges which obtain zero weight.) Doing so strictly decreases the total weight of the edges in $G$, ultimately decreasing $C$, and thereby contradicting the optimality of $Z^*$.

In fact, $G(Z^*)$ is a special kind of DAG:

Proposition 7.10. $G(Z^*)$ is a multi-tree, i.e., a directed, acyclic graph in which there is at most one directed path between any two vertices.

Proof. The proof is by contradiction, and is similar to the previous one. Suppose that $G(Z^*)$ is optimal, but there are two distinct paths, $P$ and $P'$, from some vertex $u$ to a vertex $v$. Recalling our assumption (w.l.g.) that shortest paths are unique, suppose $P$ is strictly shorter than $P'$. Let $\mathcal{C}$ denote a cycle which traverses $P$ in the forward direction, and traverses $P'$ in the reverse direction. Let $\mathcal{C}'$ denote the corresponding...
cycle on roads in $R$. Defining $g$ as before, $Z^* + g$ is feasible, and we can obtain $G(Z^* + g)$ from $G(Z^*)$ by incrementing the edge weights on $P$ and decrementing the edge weights on $P'$. This action decreases $C$, contradicting the optimality of $Z^*$. □

7.4.4 Complexity Analysis

The analysis of the prequel suggests a minimal matching algorithm in three fundamental steps.

1. Transcribe the instance of Problem 7.1 generated by $R$, $S$, and $T$;
2. Obtain an optimal integer solution $Z^*$;
3. Use the solution vector $Z^*$ to construct a matching $A^*$ with cost $C(A^*) = C(Z^*), following the procedure of Section 7.4.3.

In this section we will demonstrate that all three steps can be completed within $O(M \log M)$ time, in terms of $M$.

Transcribing Problem 7.1

An instance of Problem 7.1 is specified by the vertex supplies $b$ and the cost functions $C$, which we must therefore compute. We rely on the key fact of Prop. 7.11 below.

**Proposition 7.11.** There are at most $2m + 2M$ total linear pieces among the objectives $C(\cdot; r)$.

**Proof.** If a road $r \in R$ has no points, then $F(\cdot; r)$ equals 0 everywhere on $r$, i.e., $F(\cdot; r)$ takes values from the set $\{0\}$, and so $C(\cdot; r)$ is linear over each of the intervals $(-\infty, 0)$ and $(0, +\infty)$. The $m = |R|$ total roads result in $2m$ such “free” intervals among all roads. Any point added to a road $r$ may introduce at most one additional value in the value set of $F(\cdot; r)$. If it does, then one of the pieces of $C(\cdot; r)$ is split into two. There are at most $2M$ points, so the lemma holds. The bound can be realized as long as $|R| \geq 2$, by placing all of $S$ on one road and all of $T$ on another. □

One way to obtain the cost data is by computing the following collection of arrays, one per road $r \in R$:

$\mathcal{Y}_r = [y_0 \ldots y_{K_1+1}]$ a sorted array of the $K_1$ locations of points on $r$, and $y_0 = 0$ and $y_{K_1+1} = L_r$

$\mathcal{F}_r = [f_0 \ldots f_{K_1}]$ an array of $f_k := F(y_k; r)$ for $k = 0, \ldots, K_1$; note $F(y; r) = f_k$ for all $y \in [y_k, y_{k+1})$

$\mathcal{Z}_r = [z_0 \ldots z_{K_2}]$ a sorted array of $[z_0 = -\infty$, and] the $K_2$ consecutive integers appearing in $-\mathcal{F}_r$; notably, for all $i \in \{1 \ldots K_2\}$ and for some $j$, $z_i = -f_j$

$\mathcal{L}_r = [I_0 \ldots I_{K_2}]$ an array of “level measures” such that $I_i = \sum_{j} y_{j+1} - y_j$ where $J = \{j \mid -f_j = z_i\}$

130
\( \mathcal{C}_r = [c_0 \ldots c_{K_2}] \) an array of pairs \( c_k = (\alpha_k, \beta_k) \) such that \( C(z; r) = \alpha_k + \beta_k z \), for \( z \in [z_i, z_{i+1}] \).

For every road \( r \in R \), \( b_r \) is equal to the last entry of \( \mathcal{F}_r \), and \( \mathcal{C}_r \) contains the data of the linear pieces of \( C(\cdot; r) \) in order of appearance.

Since the road set \( R \) is known ahead of time, each point in \( S \cup T \) can be associated to a collection \( \mathcal{Y}_r \) in \( O(1) \) time (e.g., using a hash function). Therefore, sorting \( S \cup T \) into the collections \( \{\mathcal{Y}_r\} \) can be done in a single sweep of \( S \cup T \) in \( O(M \log M) \) time. Each \( \mathcal{F}_r \) can be generated by a linear scan (accumulation) over \( \mathcal{Y}_r \), and so the entire collection \( \{\mathcal{F}_r\} \) can be obtained in \( O(m + M) \) time.

\( \mathcal{Z}_r \) can be built by sorting the distinct values appearing in \( \mathcal{F}_r \), and so it takes at most \( O(m + M \log M) \) time to obtain the whole collection. \( \mathcal{L}_r \) can be built by the following procedure: Initialize the array to all zeros. Perform a scan over \( \mathcal{F}_r \). For each \( k = 0, \ldots, K_2 \), add the interval length \( y_{k+1} - y_k \) to the level measure \( I_j \) such that \( z_j = -f_k \). There are \( O(M) \) elements among all the \( \mathcal{F}_r \), so the total time to build the collection \( \{\mathcal{L}_r\} \) is \( O(m + M) \). Finally each \( \mathcal{C}_r \) can be built by a scan over \( \mathcal{L}_r \), e.g., using (7.5), and the whole collection can be obtained in \( O(m + M) \) time.

Adding up the times of all the steps, we obtain \( O(m + M \log M) \) time to compute all arrays. (The dependence on \( m \) can actually be removed, because all the arrays have implicit value for any road which obtains no points.)

**Solving Problem 7.1 for \( Z^* \)**

As demonstrated in Section 7.4.2, \( Z^* \) can be obtained in \( \log M \times O(m(m + n \log n)) \) time by a capacity-scaling algorithm, provided an \( O(1) \) procedure to compute each \( C(\cdot; r) \). One such procedure is to find the linear data \( (\alpha_k, \beta_k) \) associated with a query point \( z \), and then compute \( C(z; r) = \alpha_k + \beta_k z \). The first part can be accomplished by simple positional lookup in the array \( \mathcal{C}_r \), since the domain of the array is an ordered list of consecutive integers (i.e., \( \mathcal{Z}_r \), except \(-\infty\)). The second part is clearly \( O(1) \).

**Obtaining a Minimal Matching Given \( Z^* \)**

To obtain the final matching one must: (i) construct the graph \( G(Z^*) \), (ii) compute a topological ordering of the vertices under \( G(Z^*) \), and (iii) execute a short program (lines 5-6 of Algorithm 4) once per vertex. The graph \( G(Z^*) \) can be obtained in \( O(m + M) \) time by enumerating the intervals in the arrays \( \{\mathcal{F}_r\} \). A topological ordering of the vertices can be obtained in \( O(n + M) \) time using a standard algorithm, e.g., the one in [40, Sec. 22.4]. If the sets in the algorithm are implemented using linked-list queues, then the duties of the program per vertex are all constant time except possibly the task of splitting a list. It can be checked however, that at each of the vertices in \( S \cup T \), the split is trivial and we can skip it. In the worst case we may suffer at most \( n = |V| \) non-trivial splits, of at most \( O(M) \) operations each. Therefore, the total time of the construction is \( O(m + nM) \). Using a slightly more sophisticated data structure can reduce this time to \( O(nm + M) \), which could be significant: In a linked-list queue, any sequence of consecutive points on a single road can be represented instead using a single index range (forward or backward). Because
$G(Z^*)$ is a multi-tree, it can be proved that such queues will never obtain more than $2m$ such ranges under Algorithm 4.

**Total Runtime Complexity**

Adding the runtimes of all three components and re-arranging terms, we find that the entire algorithm can be computed in $O(M \log M + \log M \times O(m(m + n \log n))$ time. For $M$ sufficiently large compared to the description of the roadmap, the algorithm is dominated by the act of transcribing the matching instance as an instance of the convex cost flow problem, followed by construction of the matching given the optimal solution $Z^*$; in comparison to those steps, the search for the optimal solution $Z^*$ is relatively easy. Moreover, the space requirements of the algorithm are linear in $n, m, \text{and } M$, and so the algorithm is strongly polynomial.

**7.5 Discussion**

We would like to remark that the generalizations in this chapter of the techniques in [128] are essentially straightforward. Instead, the crucial contribution of the present result is recognizing that Problem 7.1 can be solved efficiently and by a strongly polynomial algorithm. Such knowledge seems to be fairly weakly disseminated; for example, as far as we know, Ahuja et. al. only published the relevant min-cost flow algorithm in their text book [4]. Depending on the sophistication of one’s standard approaches, Problem 7.1 could alternatively be reduced to a network flow problem with $O(M)$ edges, or to a generic linear optimization problem in $O(M)$ constraints. In either case, most standard contemporary approaches (algorithms) would fail to meet the quite demanding $O(M \log M)$ runtime budget: moreover, strongly polynomial algorithms tend to be elusive. While the strongly polynomial property of our algorithm stems from the fact that the solution is integral (an assignment), it might be possible to obtain a strongly polynomial, efficient algorithm for the more general transportation problem on roadmaps. This study highlights the fact that netflow flow problems are somewhat better understood, in practice, than general linear optimization problems.

Our algorithm can also provide a speed-up in cases where the roadmap geometry is implicit. Suppose, for example, that the minimal bipartite matching is desired between $S$ and $T$, where matches have costs equal to the lengths of shortest paths on a weighted, undirected graph $(S \cup T \cup V, E)$ with $n$ vertices and $m$ edges. A standard approach is to cast such problem as a minimum cost flow problem with unit supplies, which can be solved in $O(m(m + n \log n))$ time. If $m$ is very close to $2M$, then such approach represents a factor nearly $M$ speed-up, compared, e.g., to the Hungarian method on a metric completion graph. If additionally most of the $m$ edges have degree 2, then using the technique of this chapter, an additional factor nearly $M$ speed up may be possible: Letting $M'$ denote (roughly) half the number of degree-2 vertices, one can produce an equivalent matching instance of $2M'$ points on a roadmap with $n'$ vertices and $m'$ edges, where $n'$ is the number of non-degree-2
vertices, and $m'$ is their total degree. Then using the algorithm of this chapter, the instance can be solved in $O(M' \log M') + \log M' \times O(m'(m' + n' \log n'))$ time.
Chapter 8

An Explicit Formula for the Earth Mover's Distance with Continuous Road Map Distances

8.1 Introduction

The Earth Mover's distance (EMD) has played a critical role in the main results of most previous chapters. In particular, Chapter 3 demonstrated that the equations governing the performance of taxi routing policies can be extremely sensitive to the EMD. (It also appears in related “rebalancing” problems, e.g., [92].)

Recall that the EMD is a measure of distance between probability distributions, or measures more generally. The EMD between measures $\mu^x$ and $\mu^y$ in a space $(\Omega, \mathcal{D})$ is formally given by

$$W(\mu^x, \mu^y) = \inf_{\gamma \in \Gamma(\mu^x, \mu^y)} \int_{\Omega} \mathcal{D}(\mathbf{p}, \mathbf{p}') \, d\gamma(\mathbf{p}, \mathbf{p}')$$

where $\Gamma$ is the set of couplings of $\mu^x$ and $\mu^y$. The EMD is commonly interpreted as the minimum total cost of transforming one measure (“pile of dirt”) into the other, where moving $dm$ infinitesimal measure a distance $x$ incurs $x \, dm$ cost.

Unfortunately, it can be challenging to compute a reasonably precise approximation of the EMD in Euclidean $\mathbb{R}^d$ for $d \geq 2$. Interestingly, however, there is an explicit formula for the EMD on the real line, i.e., Euclidean $\mathbb{R}^1$. (We will discuss this result shortly, and formally define what it means to be “explicit” in Section 8.2.) The present chapter is motivated by a fundamental similarity between the road networks introduced in Chapter 6 and the real line segments. The former workspaces, which can be used to model environments with “street rules”, are almost everywhere locally like the latter, except that they may have a more general, “graph-like” topology. Our objective is to see if such similarity can be leveraged effectively for road networks.

Recall from Section 2.5 that when $\Omega$ is a finite set, the EMD can be expressed as the solution to a finite dimensional, linear, Network Optimization problem. Such formulation is also explicit in the sense we will define shortly. Unfortunately, if
the ground domain $\Omega$ is not finite, then explicit formulations of the EMD are only known in a few special cases, although it is usually straightforward to obtain a $1 + O(\epsilon)$ approximation, for fixed $\epsilon$, in polynomial time. (If $\Omega$ is not finite, but both distributions have finite support, then $\Omega$ can be restricted to a finite set appropriately.) The finite case has received by far the most attention in recent years, as progress on the general problem has stagnated. Indeed, the term “Earth Mover’s distance” seems to have been coined in [100] by researchers studying the discrete case, so the assumption of discrete domains is often implicit to its usage.

One of the only known non-discrete cases with an explicit formula is if $\Omega = \mathbb{R}$ and $\mathcal{P}(x, y) = |x - y|$. Then

$$W(\mu^i, \mu^b) = \int |F^i(y) - F^b(y)| \, dy, \quad (8.2)$$

where $F^i$ denotes the distribution function (d.f.) of a measure $\mu^i$, i.e., $F^i(y) := \mu^i(\{Y \in \mathbb{R} : Y \leq y\})$. (If $\mu^i$ and $\mu^b$ are probability distributions, then $F^i$ and $F^b$ are their respective cumulative density functions.) Rüschendorf discusses a few other “explicit” expressions in [102]; however, as far as we are aware, the state-of-the-art has not improved significantly since the 1980s. There is no known explicit formula for the planar ($\mathbb{R}^2$) Euclidean distances metric.

**Contributions:** The main contribution of this chapter is an explicit formulation of the Earth mover’s distance (EMD) $W(\mu^i, \mu^b)$ when the ground metric is supplied by an arbitrary, fixed road network $\mathcal{R}$. The result generalizes the formulation of the EMD in Euclidean $\mathbb{R}^1$, which (i) is the most trivial kind of road network, and (ii) had remained one of the only EMDs in a continuous domain with an explicit formula. We find that even given quite general distributions, e.g., those admitting density functions, our formulation casts the EMD as the optimal value of a finite-dimensional, real-valued optimization problem with a convex objective function and linear constraints, which is highly amenable to convex programming techniques [21]. In the special case that the distributions $\mu^i$ and $\mu^b$ have piece-wise constant density, the problem reduces to one whose objective function is convex quadratic, in number of variables linear in the number of pieces. One can solve such a problem efficiently using standard quadratic programming (QP) methods.

**Organization:** This chapter in particular requires much of the background on geometry, network flow theory, and the Earth Mover’s distance, presented in Sections 2.1.3, 2.4, and 2.5 respectively. We also rely on the same mathematical formalism for road networks (as metric spaces), introduced in Section 6.2.

The rest of the chapter is organized as follows. First, we state formally the objectives of the chapter in Section 8.2. In Section 8.3, we present the main result of the chapter, an explicit formulation of the EMD on road networks as a finite-dimensional convex optimization problem. In Section 8.4 we present a numerical study to illustrate the theory with practical intuition. In Section 8.5, we provide a naive, general-purpose procedure to compute an approximation of the EMD for any ground metric. In Section 8.6, we refine the procedure using structural knowledge about road networks to obtain a procedure which is simultaneously more efficient and
more insightful. (These approximations are integral components to a formal proof of the correctness of our main result, presented later in the chapter.) In Section 8.7 we analyze the computational space and runtime complexity of the procedures of Sections 8.3, 8.5, and 8.6. We provide the formal proof of correctness of our main result in Section 8.8. Finally, we present concluding remarks in Section 8.9.

8.2 Problem Statement

For the rest of the chapter, we will say that a formula is explicit if it is a closed-form expression or an integral involving closed-forms, or if it is a convex program in terms of such expressions for which strong duality holds [27, Ch. 5]. It is essentially straightforward to compute such formulas, because closed-forms are “well-studied”, and efficient techniques exist both for numerical integration and convex optimization [27, Ch. 11]. Many of the distributions on \( \mathbb{R} \) which are commonly used to represent other ones have cdfs which are considered closed-form. Network optimization problems [21, Ch. 5] are among a broad class of convex optimization problems satisfying strong duality.

The objective of the chapter is to obtain an explicit formulation of the Earth Mover’s distance, given a roadmap \( R \), as a network optimization problem. We will consider a fairly general class of measures satisfying the following technical condition.

**Assumption 8.1.** We restrict our attention to finite, absolutely continuous probability distributions (unity total measure) on road networks, with Lipschitz road densities.

In this chapter, we will denote by \( \varphi^1 = \{\varphi^1_r\}_{r \in R} \) and \( \varphi^2 = \{\varphi^2_r\}_{r \in R} \) the densities of distributions \( \mu^1 \) and \( \mu^2 \), respectively.

8.3 The Earth Movers Distance on Road Networks

8.3.1 Main Result—One-dimensional Earth mover’s distance

The main result of the chapter is Theorem 8.1 below:

**Theorem 8.1 (One-dimensional Earth mover’s distance).** For any road network environment \( (R, \mathcal{D}) \) and any two measures \( \mu^1 \) and \( \mu^2 \) over \( R \), satisfying Assumption 8.1 with equal total (finite) measure, let \( F^2_r(\cdot) \equiv \Phi(\cdot; \varphi^2_r) \) and \( F^1_r(\cdot) \equiv \Phi(\cdot; \varphi^1_r) \) for every road \( r \in R \), with \( \Phi(\cdot; \varphi) \) defined as in (6.2). Then

\[
\mathcal{W}(\mu^1, \mu^2) = \min \left\{ \sum_{r \in R} \int_0^{l_r} \left| x_r + F^2_r(y; r) - F^1_r(y; r) \right| \, dy : x \in \mathbb{R}^R, \sum_{r: r \to u} x_r = \sum_{r: r \leftarrow u} x_r + \mu^1(r) - \mu^2(r) \text{ for all } u \in V \right\}.
\]

(8.3)
(8.3) is a general extension of the one-dimensional result (8.2) and can be written as a finite-dimensional network optimization problem. It is easy to check that (8.3) reduces to (8.2) when the roadmap is a single segment.

The proof of Theorem 8.1 requires many intermediate results that we have not yet established. Moreover, the analysis of the chapter will proceed in terms of an equivalent technical formulation, which we will develop starting in the sequel. We omit the proof of Theorem 8.1, but obtain the result by the equivalence between the two formulations, which we prove later on.

8.3.2 Alternative (Technical) Formulation

In this section we provide a procedure to construct an alternative but equivalent instance of the Network Optimization problem, whose objective is convex and whose (optimal) solution has cost equal to \( W(\mu^a, \mu^b) \). An instance of the Network Optimization problem is the pair \((\mathcal{N}, c)\) of a flow network \(\mathcal{N}\) and edge costs \(c\) (see Section 2.4).

For the alternative formulation of the EMD, we make two additional simplifying assumptions.

Technical Assumptions

**Assumption 8.2.** For technical reasons, we assume that the supports of \(\mu^a\) and \(\mu^b\) are disjoint; e.g., it holds that \(\varphi^a_r(y) \times \varphi^b_r(y) = 0\) for all \(r \in R, y \in [0, L_r]\).

Assumption 8.2 is actually without loss of generality, since one may subtract the min of \(\mu^a\) and \(\mu^b\) without altering the EMD (Corollary 2.19, generalized).

**Assumption 8.3.** Let \(\mu(r)\) denote the total measure of road \(r\) under distribution \(\mu\). We assume that \(\mu^a(r) \times \mu^b(r) = 0\) for all \(r \in R\); that is, only one of the input distributions may be positive on any given road.

Assumption 8.3 supercedes Assumption 8.2, but it is also quite benign. Roads satisfying Assumptions 8.1 and 8.2 but not 8.3 can be “cracked”—by injecting additional vertices—such that Assumption 8.3 becomes satisfied. Such insertions do not alter the essential structure of the road network, e.g., shortest-path distances are preserved.

For any road \(r\), if \(\mu^a(r) > 0\), then we call it a supply road; if \(\mu^b(r) > 0\), then we call it a demand road. According to Assumption 8.3, a road may be either a supply road or a demand road, but not both; if it is neither, i.e., \(\mu^a(r) = \mu^b(r) = 0\), then we call it a transshipment road.

We can write the set of supply roads as \(S := \{r \in R : b(r) > 0\}\), demand roads as \(D := \{r \in R : b(r) < 0\}\), and transshipment roads as \(T := \{r \in R : b(r) = 0\}\).

Preliminary Discussion

Unfortunately, our construction is quite technical. Therefore, first, we provide an informal outline of our approach, motivated by the “pile of dirt” analogy:

Consider a single road \(r\) of length \(L\) (see Figure 8-1). Suppose that \(r\) is a supply
Figure 8-1: A supply road \( r \in S \). The area \( x \) (under the curve to the left of \( y^* \)) is transported to \( r^- \). The area \( \mu^+(r) - x \) (to the right of \( y^* \)) is transported to \( r^+ \).

road, whose distribution of commodity has density function \( \varphi \). Since \( r \) is a supply road, all of the demand is elsewhere in the network. Therefore, all the available commodity must leave \( r \) by one of its endpoints. Suppose we wish to transport a quantity \( x \) of the commodity via \( r^- \), and the remaining \( \mu^+(r) - x \) commodity via \( r^+ \). If the cost of transportation is proportional to distance traveled, it is easy to argue that moving the left-most \( x \) commodity to \( r^- \) and the remainder to \( r^+ \) is optimal (see Figure 8-1). The boundary separating the left-most \( x \) commodity from the remainder lies at coordinate \( y^* := \Psi(x; \varphi) \). Applying basic calculus, the cost of this strategy is determined to be

\[
\int_0^{y^*} \varphi(y) \, y \, dy + \int_{y^*}^L \varphi(y) \, [L - y] \, dy.
\]  

A symmetrical argument obtains the same cost expression for transporting commodity into a demand road. Note that we have left the real-valued quantity \( x \) as one degree of freedom subject to optimization.

It may not be possible for all commodity which leaves some road by one of its endpoints to supply demand on the interior of an adjacent road. For example, if the total supply on one road exceeds the total demand of its immediate neighborhood, then some supply must be assigned elsewhere in the network. To accomplish this, let us consider a “strategy” in three phases: First, commodity will be “accumulated” at vertices as previously described. The third phase will be exactly opposite in the sense that commodity will be “dispersed” from the vertices to satisfy demand in the interiors of adjacent roads. During the middle phase, however, commodity may be “re-distributed”, but strictly on the vertex set \( V \). The problem of finding the minimum cost re-distribution schedule given the two “vertex-only” distributions of commodity (i.e., the one immediately after accumulation and the one immediately before dispersion), can be cast as a traditional minimum-cost flow problem, on the skeleton of the roadmap \( (V, R) \), using the lengths of the roads as edge weights. Fortunately, it turns out that the optimal strategy of this three-phase type is at least as good as any other strategy.
Instance Construction

We will distinguish the components of our particular instance using the \( W \) superscript. We will refer to the flow network \( N^W \) as the Wasserstein network. The construction of the network \( N^W \) is as follows:

We begin with both \( V^W \) and \( A^W \) empty. Then, we insert into \( V^W \) the whole collection of roads and interchanges \( R \cup V \). While the roads in \( R \) are edges of the roadmap, they are treated simply as vertices in \( N^W \). Let \( b(r) := \mu^r - \mu^s(r) \) be called the \textit{surplus} of road \( r \). The supplies associated with \( V^W \) will be

\[
b^W(u) := b(u) \quad \text{for} \quad u \in R; \quad 0 \quad \text{for} \quad u \in V. \quad (8.5)
\]

For each supply road \( r \in S \), we insert directed edges \( (r, r^-) \) and \( (r, r^+) \) into \( A^W \). Even in the case \( r^- = r^+ \), these notations will denote two separate and distinct edges (though, in such case, with the same endpoints); therefore, note that \( N^W \) could be a multi-graph. We will use the alias \( tconn_r \) to refer to \( (r, r^-) \) and \( hconn_r \) to refer to \( (r, r^+) \). For each demand road \( r \in D \), we add the edges \( (r^-, r) \) and \( (r^+, r) \) into \( V^W \); such edges are also always distinct, and are also given aliases \( tconn_r \) and \( hconn_r \) (respectively), though they have the opposite direction. \( A^W \) now contains the decision edges; let us denote this set \( A^{Dec} \).

The costs on the decision edges are as follows. Let

\[
\varphi_r := \begin{cases} 
\varphi^r_r, & \text{if } r \in S \\
\varphi^s_r, & \text{if } r \in D,
\end{cases}
\]

and

\[
\chi_r(x) := \varphi_r (L_r - x) \quad \text{for all } r \in S \cup D.
\]

Let

\[
q(x; \varphi) := \int_{y=0}^{\Psi(x; \varphi)} \varphi(y) \; y \; dy \quad \text{for any } \varphi.
\]

Then

\[
c^W(\cdot; tconn_r) := q(\cdot; \varphi_r), \quad \text{and} \quad (8.7)
\]

\[
c^W(\cdot; hconn_r) := q(\cdot; \chi_r), \quad \text{for all } r \in S \cup D. \quad (8.8)
\]

Note that the cost functions (8.7) and (8.8) provide the first and second terms of (8.4), respectively, of the intuitive model.

Now let \( A^{Rte} \) denote a set of routing edges: \( A^{Rte} \) contains one edge in each direction between any pair \( u, v \in V \), if \( u \neq v \) and they are connected by some \( r \in R \); such edge has linear cost with weight \( w^{Rte}((u, v)) \) equal to the length of the shortest such road. We insert all of the routing edges into \( A^W \). At this point, the flow conservation constraints (7.7) on \( N^W \) account precisely for the flow conservation requirements of all three “phases” in the intuitive approach.

Figure 8-2(a) shows a simple road network with roads “north” (N), “east” (E), “south” (S), and “west” (W), and Figure 8-2(b) shows the corresponding Wasserstein network. E and S are supply roads and N and W are demand roads; therefore, notice that the decision edges—shown by thick lines—point \textit{out} of E and S, while they point
(a) A square road network with roads: North (N), East (E), South (S), and West (W).

(b) The Wasserstein network resulting from the roadmap in Figure 8-2(a).

Figure 8-2: A simple road network and the resulting “Wasserstein” flow network.

into N and W. Each road also contributes a pair of routing edges to the network.

Main Result, Alternative Form

Theorem 8.2 (Main Result (Alternative Form)). For any road network environment \((\mathcal{R}, \mathcal{D})\), described by a roadmap \((V, R)\), and any two measures \(\mu^z\) and \(\mu^\lambda\) over \(\mathcal{R}\), satisfying Assumptions 8.1 and 8.3 with equal total (finite) measure,

\[
\min_{f \in \mathcal{N}^w} J(f; c^w) = \min_{f \in \mathcal{N}^w} \sum_{a \in A^w} c^w (f(a); a) = \mathcal{W}(\mu^z, \mu^\lambda).
\]

(8.9)

Note that the left-hand side of (8.9) is a finite-dimensional optimization problem with only linear equality and inequality constraints. The proof of Theorem 8.2 also requires a number of intermediate results that we have not yet established, so we defer the proof until the end of the chapter.

8.3.3 Convexity of the EMD Objective

The next result is crucial to show that the formulation of Theorem 8.2 is explicit.

Theorem 8.3. The objective function \(J(\cdot; c^w)\) is convex over \(f \in \mathcal{N}^w\).

Proof. Theorem 8.3 follows as an easy consequence of the next proposition, since sums of convex functions are convex.

Proposition 8.4. For every Lipschitz function \(\varphi : [0, L] \to \mathbb{R}_{\geq 0}\), \(q(\cdot; \varphi)\) is Lipschitz continuous and convex over the interval \([0, \Phi(L; \varphi)]\).

Proof. The absolute difference \(|q(x'; \varphi) - q(x; \varphi)|\) can be written as

\[
|q(x'; \varphi) - q(x; \varphi)| = \left| \int_{\Psi(x'; \varphi)} \varphi(y) \, dy \right|.
\]
Because the range of $\Psi(\cdot; \varphi)$ is $[0, L]$, we have $y \leq L$ over the whole integral range. Therefore,
\[
\left| \int_{\Psi(x_0; \varphi)}^{\Psi(x'; \varphi)} \varphi(y) \, dy \right| \leq L \left| \int_{\Psi(x_0; \varphi)}^{\Psi(x'; \varphi)} \varphi(y) \, dy \right| = L |x' - x|.
\]

Then $q(\cdot; \varphi)$ is Lipschitz, because for every $x, x' \in [0, \Phi(L; \varphi)]$
\[
\left| \frac{q(x'; \varphi) - q(x; \varphi)}{|x' - x|} \right| \leq L.
\]

To show that $q(\cdot; \varphi)$ is convex, we observe that for all $x_0, x \in [0, \Phi(L; \varphi)]$ it holds
\[
q(x; \varphi) \geq q(x_0; \varphi) + \Psi(x_0; \varphi)[x - x_0], \tag{8.10}
\]
i.e., there is a tangent line at every point $x_0 \in [0, \Phi(L; \varphi)]$ with $q(\cdot; \varphi)$ lying entirely above it. Such functions are known to be convex, e.g., by convexity of epigraphs which can be expressed as the intersection of many linear epigraphs. To verify (8.10), one can write
\[
q(x; \varphi) - q(x_0; \varphi) = \int_{\Psi(x_0; \varphi)}^{\Psi(x; \varphi)} \varphi(y) \, dy
\]
\[
\geq \Psi(x_0; \varphi) \int_{\Psi(x_0; \varphi)}^{\Psi(x; \varphi)} \varphi(y) \, dy = \Psi(x_0; \varphi)[x - x_0].
\]

While $q$ may be difficult to obtain in analytical form, except in special cases, (8.10) demonstrates that $\Psi$ is everywhere in its subgradient. Gradient and subgradient methods are at the heart of modern algorithms for constrained optimization of general convex functions, and Theorem 8.3 provides a certificate that $q(\cdot; \varphi)$ is convex regardless of the density function $\varphi$. Therefore, provided one has access to an evaluable expression (or “circuit”) for $\Psi$, then our formulation is highly amenable to modern convex optimization techniques.

Road-wise Uniform Density

In the special case that all of the road densities are uniform, then we obtain $\Psi(x; \varphi) = x/\rho$ and $q(x; \varphi_r) = \frac{1}{2} x^2 / \rho$ for each $r \in R$, where $\rho$ is the uniform density of $\varphi$. Thus, if the density functions are uniform over all segments, then the decision edge costs are all convex quadratic in $f$. The resulting class of network optimization problems can be solved by way of quadratic programming (QP), a well-studied approach to optimization problems with convex quadratic objective and linear constraints [27, p.152].
8.3.4 Equivalence of Two Formulations

To finish the section, we provide a proof of the equivalence between the two formulations (8.3) and (8.9), by which (after proving Theorem 8.2) we obtain the result of Theorem 8.1.

Proposition 8.5. (8.3) and (8.9) are equivalent.

Proof. We assume that Assumption 8.3 holds without loss of generality. If it does not, then the proof will follow after an appropriate transformation of the roadmap. The equivalence is through the relation $x_r = f(r^-, r) - f(r, r^-)$, where $f(r^-, r) = f(t_{\text{conn}, r})$ if $r \in D$ and zero otherwise, and $f(r, r^-) = f(t_{\text{conn}, r})$ if $r \in S$ and zero otherwise. (If $r$ is the designated shortest road between $r^-$ and $r^+$, then $x_r = [f(r^-, r) - f(r, r^-)] + [f((r^-, r^+)) - f((r^+, r^-))];$ however, we handle that case next.) It is straightforward to check that this relation provides a mapping between the conservation constraints of (8.9) and those of (8.3).

In the case that $r \in D$ (i.e., $\mu^r(r) = 0$), then $x_r = f(r^-, r)$, and the contribution of (8.3) on $r$ equals

$$\int \left| x_r - F^y(y; r) \right| \, dy = \int_0^{\Psi(x_r; \varphi^r_y)} \left[x_r - F^y(y; r)\right] \, dy - \int_{\Psi(x_r; \varphi^r_y)}^{L_r} \left[x_r - F^y(y; r)\right] \, dy. \tag{8.11}$$

Performing integration by parts with $u = x_r - F^y(y; r)$ and $dv = dy$, we obtain

$$uv - \int v \, du = y \left[ x_r - F^y(y; r) \right] + \int y \varphi^y_r(y) \, dy. \tag{8.12}$$

Substituting (8.12) in (8.11), and cancelling terms due to $x_r = F^y(\Psi(x_r; \varphi^r_y); r)$, we obtain

$$\int \left| x_r - F^y(y; r) \right| \, dy = \int_0^{\Psi(x_r; \varphi^r_y)} y \varphi^y_r(y) \, dy - L_r \left[ x_r - F^y(L_r; r) \right] - \int_{\Psi(x_r; \varphi^r_y)}^{L_r} y \varphi^y_r(y) \, dy. \tag{8.13}$$

Finally, substituting

$$x_r - F^y(L_r; r) = -\int_{\Psi(x_r; \varphi^r_y)}^{L_r} \varphi^y_r(y) \, dy$$

above, we may write

$$\int \left| x_r - F^y(y; r) \right| \, dy = \int_0^{\Psi(x_r; \varphi^r_y)} y \varphi^y_r(y) \, dy + \int_{\Psi(x_r; \varphi^r_y)}^{L_r} [L_r - y] \varphi^y_r(y) \, dy. \tag{8.13}$$

Recalling $x_r = f(r^-, r)$, the right-hand side of (8.13) is precisely $q(f(r^-, r); \varphi^y_r) + q(\mu^r(r) - f(r^-, r); \chi^y_r)$, which is the contribution of the decision edges associated with $r$ in (8.9). In the case that $r$ is the shortest road between $r^-$ and $r^+$, a similar
Figure 8-3: The roadmap of Figure 8-2(a) labeled with measures $\mu^y$ and $\mu^b$.

\[ \begin{align*}
\mu^b(N) &= \frac{1}{5} \\
\mu^b(W) &= \frac{4}{5} \\
\mu^b(E) &= \frac{2}{5} \\
\mu^s(S) &= \frac{3}{5}
\end{align*} \]

Figure 8-4: Wasserstein network of the measures in Figure 8-3, labeled with the optimal flow and induced costs.

(a) Optimal flow under $\mu^y$ and $\mu^b$

(b) Cost per-edge of optimal flow

**Figure 8-4**: Wasserstein network of the measures in Figure 8-3, labeled with the optimal flow and induced costs.

derivation will include also the contribution of the routing edges generated by $r$, i.e.,

\[ \int |x_r - F^b(y; r)| \ dy = q(f(r^-, r); \varphi^r_r) + q(\mu^y(r) - f(r^-, r); \chi^y_r) + L_r f((r^-, r^+)). \]

(An important part of that derivation is recognizing that $f((r^-, r^+)) > 0 \implies f(r^-, r) = \mu^y(r).$) A symmetrical procedure produces symmetrical results for $r \in S$, and collecting all of the cost terms proves the lemma.

8.4 A Numerical Example

Let us re-visit the example network in Figure 8-2(a) and assign specific distributions. Suppose each road is of unit length, and has probability given in Figure 8-3. The supply or demand of each road is distributed uniformly over its length. (Actually, these two distributions are precisely the factor distributions (marginals) of the joint distribution examined in Section 6.7; e.g., see Table 6.3.)

Figure 8-4 shows two new copies of the flow network $\mathcal{N}^{\text{xy}}$ first shown in Figure 8-2(b). The network in Figure 8-4(a) is labeled with the flows of the optimal network flow solution (obtained by solving a quadratic program). The network in Figure 8-4(b) is labeled with the costs incurred on each edge by the optimal network flow.
The optimal solution has cost equal to $31/30$, which is therefore the Earth Mover’s distance between $\mu^\pi$ and $\mu^\nu$.

Examining the optimal flow provides qualitative insight in addition to the value of the EMD. In particular, we can observe the following facts: First, the demand of the north road (N) is supplied entirely by the east road (E). Second, all of the supply of the south road (S) goes to the west road (W). Finally, the east road (E) supplies the remaining demand of the west road (W), however, $1/15$ unit of supply from E reaches W via the clockwise path (E-3-4-W), while the remaining $2/15$ unit of supply reaches W via the counter clockwise path (E-2-1-W).

8.5 Approximating the Earth Movers Distance by Min-Cost Flow

The rest of the chapter explores a particular method to prove Theorem 8.2, i.e., the correctness of our algorithm. At a high-level, our approach is to develop an approximation scheme for $\mathcal{W}$ (the EMD), bounding it entirely between an inner- and outer-approximation, and then showing that the bounds converge (squeeze) to the LHS of (8.9).

8.5.1 The General Purpose Scheme

In this section we present a naive, “general-purpose” approximation scheme for the Earth Movers distance for a fairly general class of metric domains. Specifically, we present a procedure which, given a particular partition $\mathcal{C}$ and argument distributions $\mu^\pi$ and $\mu^\nu$, generates a matched pair of network optimization problem instances. The optimal solutions to these instances will bound $\mathcal{W}(\mu^\pi, \mu^\nu)$ from both sides. If one can obtain a tessellation scheme for the domain $\Omega$, capable of tessellating any compact workspace $\mathcal{Q} \subset \Omega$ to increasingly high “resolution”, then $\mathcal{W}$ can be approximated by making such bounds arbitrarily close. (Such tessellation is easily obtainable, e.g., in Euclidean environments.)

**Workspace Tessellation**: The ability to tessellate is generally a property specific to the type of the domain $\Omega$. A common tessellation scheme for Euclidean $\mathbb{R}^d$ is the grid-based partition of $\mathbb{R}^d$ into [hyper-] cubic cells of side-length $\frac{1}{2}d^{-1/2}$. The key objective of tessellation in this chapter is to ensure that for any $\epsilon > 0$ one can produce a partition $\mathcal{C}_\epsilon$ satisfying

$$\max_{p \in C, p' \in C'} \mathcal{D}(p, p') - \min_{p \in C, p' \in C'} \mathcal{D}(p, p') \leq \epsilon \quad \text{for all } (C, C') \in \mathcal{C}_\epsilon^2. \tag{8.14}$$

**Instance Construction**: Let $\mathcal{C}$ be a finite partition of a workspace $\mathcal{Q} \in \Omega$. The flow network $\mathcal{N}^{\text{APPROX}}$ will comprise a di-graph $(\mathcal{V}^{\text{APPROX}}, \mathcal{A}^{\text{APPROX}})$ and supplies $b^{\text{APPROX}}$. We will call $\mathcal{N}^{\text{APPROX}}$ the approximation network. To construct the vertex set $\mathcal{V}^{\text{APPROX}}$ we generate two sets $V^x$ and $V^y$ of new symbolic vertices; each set is of cardinality $|\mathcal{C}|$. We assign two such vertices to each cell $C \in \mathcal{C}$, one from the set $V^x$ and one from the set $V^y$, where each vertex is assigned to a single cell only.
(see Figure 8-5). Let bipartite matchings $M^z$ (between $V^z$ and $C$) and $M^p$ (between $V^p$ and $C$) denote the respective assignments. (For example, if $u$ is the vertex in $V^z$ assigned to $C \in C$, then $(u, C) \in M^z$.) We define the supplies as

$$b_{\text{APPROX}}(u) := \mu^z(C) \quad ((u, C) \in M^z), \quad (8.15)$$

$$b_{\text{APPROX}}(v) := -\mu^p(C) \quad ((v, C) \in M^p). \quad (8.16)$$

Let $A_{\text{APPROX}}$ form the complete bipartite graph between $V^z$ and $V^p$, i.e., $A_{\text{APPROX}} := V^z \times V^p$. Let $w_{\text{LOWER}} := \{\overline{w}_{(u,v)}\}$ be the set of edge weights on $A_{\text{APPROX}}$ satisfying

$$\overline{w}_{(u,v)} = \min_{p \in C, p' \in C'} \|p, p'\| \quad \text{for} \quad (u, C) \in M^z, (v, C') \in M^p, \quad (8.17)$$

and let $w_{\text{UPPER}} := \{\overline{w}_{(u,v)}\}$ be the set of edge weights satisfying

$$\overline{w}_{(u,v)} = \max_{p \in C, p' \in C'} \|p, p'\| \quad \text{for} \quad (u, C) \in M^z, (v, C') \in M^p. \quad (8.18)$$

### 8.5.2 Approximation Bounds

The network $N_{\text{APPROX}}$ captures a hypothetical scenario (by aggregation of points into a finite number of cells) where the cost of transportation (distance) from one cell to another is a single constant regardless of the particular endpoints. The costs $c_{\text{LOWER}}$ are "optimistic", assigning cost to a pair of cells equal to the minimum distance between endpoints in either cell; the costs $c_{\text{UPPER}}$, meanwhile, are "pessimistic", assigning cost equal to the maximum such distance. As the fine-ness of the tessellation increases, in the sense that $\epsilon \to 0^+$ in (8.14), the difference between the optimistic and pessimistic costs will vanish. Such intuition supports the claims of Propositions 8.6 and 8.7, below; the formal proofs, however, are provided in Appendix C.1.

**Proposition 8.6.** For any distributions $\mu^z$ and $\mu^p$ satisfying Assumptions 8.1 and 8.3, any $\epsilon > 0$, and any partition $C_\epsilon$ of workspace $Q \subset \Omega$ satisfying (8.14), let $N_{\text{APPROX}}$ denote the approximation network of Section 8.5.1 having weights $w_{\text{LOWER}}$
and \( w_{\text{UPPER}} \). Let

\[
\mathbb{W} = \min_{f \in \mathcal{N}_{\text{APPROX}}} J(f; w_{\text{LOWER}}) \\
\overline{\mathbb{W}} = \min_{f \in \mathcal{N}_{\text{APPROX}}} J(f; w_{\text{UPPER}}).
\]

Then \( \mathbb{W} \leq \mathbb{W}(\mu^2, \mu^\ast) \leq \overline{\mathbb{W}} \).

**Proposition 8.7.** Under the same condition as Proposition 8.6, \( \overline{\mathbb{W}} - \mathbb{W} \leq \varepsilon |\mu| \), where \( |\mu| \) denotes the constant total measure of either \( \mu^2 \) or \( \mu^\ast \).

Together, Propositions 8.6 and 8.7 prove that \( \mathbb{W} \rightarrow \mathbb{W}(\mu^2, \mu^\ast)^- \) and \( \overline{\mathbb{W}} \rightarrow \mathbb{W}(\mu^2, \mu^\ast)^+ \) as \( \varepsilon \rightarrow 0^+ \), i.e., both converge to \( \mathbb{W}(\mu^2, \mu^\ast) \).

## 8.6 Approximating the EMD on Road Networks

### 8.6.1 The General-Purpose Scheme

Road networks are sufficiently like Euclidean \( \mathbb{R}^1 \) that a small modification to the grid-based tessellation scheme of Section 8.5.1 obtains the same convergence in the approximation by \( \mathcal{N}_{\text{APPROX}} \) as the grid-based scheme does for \( \mathbb{R}^d \): For each \( r \in \mathcal{R} \), let \( N_r := \lfloor L_r / \varepsilon \rfloor \) and let \( \varepsilon_r := L_r / N_r \). Then one can partition each road \( r \in \mathcal{R} \) into \( N_r \) segments of length \( \varepsilon_r \). We will refer to such partition as the \( \varepsilon \)-tessellation of \( \mathcal{R} \). The interval lengths \( \{\varepsilon_r\}_{r \in \mathcal{R}} \) are all smaller than \( \varepsilon \), so the resulting partition satisfies (8.14) and Propositions 8.6 and 8.7 hold.

While our pain-staking attention to network flow-based approximation schemes may be misleadingly algorithmic, our interest in them is not to approximate \( \mathbb{W} \), but to discover a sequence \( \mathbb{W}_k \) which converges to \( \mathbb{W} \) and has an analytical limit. Unfortunately, the network structure generated by the general-purpose scheme is too general to reveal any underlying analytical form of \( \mathbb{W} \). Fortunately, that scheme is not the only network flow-based approximation scheme that we may use.

### 8.6.2 The Path-based Scheme

In this section, we present another approximation scheme which leverages the structure of the road network \( \mathcal{R} \). We will call our alternative approximation scheme the “path-based” scheme. An important feature of the scheme is that it uses the same \( \varepsilon \)-tessellation of \( \mathcal{R} \), and many of the same network vertices (i.e., \( V_{\text{APPROX}} \)), as the general-purpose one. The scheme differs in that we seek an alternative flow network topology. Our goal is to obtain additional insight into computing the EMD. Naturally, the new scheme must preserve the cost of the min-cost flow. (Because either of the squeezing bounds converges to \( \mathbb{W} \), we focus only on the lower bound produced by \( c_{\text{LOWER}} \).)

The ability to produce a meaningful alternative topology is based on two important observations about network flows: First, while network flows are most commonly represented as mappings from individual edges to flow volume, they can be represented
equally well by mapping from *paths* to flow volume. For example, the network flow in Figure 8-4(a) can be interpreted as a so-called “path and cycle flow”, with 1/5 unit flow on the path (E-2-N), 2/15 flow on the path (E-2-1-W), 1/15 flow on the path (E-3-4-W), and 3/5 flow on the path (S-4-W). The second observation is that in the absence of edge “capacities” (which do not arise in this chapter), minimum-cost network flows are supported *entirely* on shortest paths.

**Definition 8.8 (Path and cycle flows).** Let $\mathcal{P}$ denote the set of simple paths on a (multi-)digraph $G = (V, A)$, and let $\mathcal{Q}$ denote the set of cycles. A path and cycle flow is a mapping $f : \mathcal{P} \cup \mathcal{Q} \to \mathbb{R}_{\geq 0}$. (We will call flows of the former type $(A \to \mathbb{R}_{\geq 0})$ arc flows, or simply flows.)

Path and cycle flows determine arc flows in a natural way, such that the flow on an edge is equal to the sum of all flows on paths and cycles that use the edge. Defining the delta function $\delta_a(P)$ for each $a \in A$—equal to 1 if $a$ is included in the path or cycle $P \in \mathcal{P} \cup \mathcal{Q}$, and 0 otherwise—then the arc flow $\hat{f}$ described by a path and cycle flow $f$ is determined by

$$\hat{f}(a) = \sum_{P \in \mathcal{P} \cup \mathcal{Q}} \delta_a(P)f(P) \quad \text{for all } a \in A. \quad (8.21)$$

A path and cycle flow is admissible if its arc flow is admissible. Letting $|P|_w$ denote the total weight of a path $P$ on a weighted network $(\mathcal{N}, w)$, i.e., $|P|_w = \sum_{a \in A} \delta_a(P)w(a)$, the cost of a path-and-cycle flow can be written $J(f; w) = \sum_{P \in \mathcal{P} \cup \mathcal{Q}} f(P)|P|_w$. A path-and-cycle flow has the same total weight as its arc flow.

**Lemma 8.9.** Let $(\mathcal{N}, w)$ and $(\mathcal{N}, \tilde{w})$ be two weighted flow networks satisfying the following properties:

1. Every supply vertex has the same supply in $\mathcal{N}$ and $\tilde{\mathcal{N}}$;
2. Every demand vertex has the same demand in $\mathcal{N}$ and $\tilde{\mathcal{N}}$;
3. The total weight of the weighted shortest path, from any supply vertex to any demand vertex, is the same in both networks.

Let $J^*$ and $\tilde{J}^*$ denote the costs of the minimum-cost flows on $\mathcal{N}$ and $\tilde{\mathcal{N}}$, respectively (and with respective weights). Then $J^*$ and $\tilde{J}^*$ are equal.

By Lemma 8.9, it is possible to substitute an alternative topology over the network vertices $V^{\text{APPROX}}$, without changing the value of the minimum cost flow, so long as every shortest path from a supply vertex $u$ to a demand vertex $v$ has length equal to the weight of edge $(u, v)$ in $\mathcal{N}^{\text{APPROX}}$. Our proof of the lemma requires elements of the next Theorem, reproduced from [4]:

**Theorem 8.10 (Theorem 3.5 of [4] (annotated)).** Every path and cycle flow has a unique representation as nonnegative arc flows [i.e., (8.21)]. Conversely, every nonnegative arc flow can be represented as a path and cycle flow (though not necessarily uniquely) with the following two properties:
1. Every directed path with positive flow connects a [supply] node to [a demand] node.

2. (not needed for our discussion, see [4] for full text).

Proof of Lemma 8.9. It is sufficient to prove $\tilde{J}^* \leq J^*$, since the two networks commute in the statement of the lemma. Let $f^*$ be the path-and-cycle representation of the minimum-cost flow on $N$. By Property 1 of Theorem 8.10, every positive-flow path is from a supply node to a demand node. Each positive-flow path is also a shortest path (this can be proved by a simple substitution argument). We can construct a path-and-cycle flow $\tilde{f}$ on $\tilde{N}$ by adding the weight of each positive-flow path in $f^*$ into $\tilde{f}$ on the shortest directed path between the same endpoints. Properties 1 and 2 of Lemma 8.9 ensure that $\tilde{f} \in \tilde{N}$ (it is admissible). By Property 3, the latter paths have the same weight as the former ones, proving the total cost of $\tilde{f}$ is the same as that of $f^*$. $\tilde{J}^*$, by definition, cannot be more. □

Instance Construction: Our construction must satisfy Lemma 8.9 with $N^\text{approx}$. Note that Properties 1 and 2 are quite easy to satisfy, i.e., by letting $b^\text{PATH}$ equal $b^\text{approx}$ on $S \cup D$ and zero anywhere else. In order to satisfy Property 3, we seek to construct a network where the shortest path from $u \in V^\text{s} (u, C) \in M^\text{s}$ to $v \in V^\text{d} (v, C') \in M^\text{d}$ has total weight equal to that given by $w^\text{lower}$, or the minimum distance on $R$ from $C$ to $C'$, i.e. (8.17). The crucial observation is that any path from $C$ to $C'$ can be decomposed into three parts: (i) first, a path from $C$ to an endpoint $r^\pm$ of the road $r \in R$ for which $C \subset r$; (ii) second, a path from that endpoint $r^\pm$ to an endpoint $\tilde{r} \in R$, $C' \subset \tilde{r}$; (iii) finally, a path from the second endpoint $\tilde{r}^\pm$ to the cell $C'$.

To obtain the network $N^\text{PATH}$ instance we start with $V^\text{PATH} := V$ (the vertices of $R$) and $A^\text{PATH} := \emptyset$. Then, for each non-transshipment road $r \in S \cup D$, we insert into the graph $(V^\text{PATH}, A^\text{PATH})$ one of two possible "road devices". If $r$ is a supply road, i.e., $r \in S$, then we add a "supply device", as shown in Figure 8-6; The vertices of this device are the ones in $V^\text{s} \subset V^\text{approx}$ associated with the tessellation of $r$; as seen in Figure 8-6, they are ordered from $r^-$ to $r^+$. Otherwise, if $r$ is a demand road ($r \in D$), then we add a "demand device", which is like the supply device, except (i) the vertices are those from $V^\text{d}$, and (ii) tconn, and hconn, have the opposite direction.
(In either case, tconn_r has endpoints u_r^1 and r^-, and hconn_r has endpoints u_r^N and r^+.) We denote by g_r the device subgraph belonging to road r.

**Remark 8.11.** The resulting set $V^\text{PATH}$ is not exactly that same set as $V^\text{APPROX}$. We observe, however, that the symmetric difference set includes only non-supply, non-demand vertices, which cannot contribute positive flow paths to a minimum-cost flow; thus, they do not affect compliance with Lemma 8.9.

As indicated in Figure 8-6, let the weights $w^\text{PATH}$ give $r_c = L_r/n$ on all the road device edges except tconn_r and hconn_r which are "free" (zero cost). Such weights are carefully chosen to ensure that: (i) the shortest path from $u \in g_r$ to either endpoint $r^\pm$ has total weight equal to the distance on $R$ from $C$ to $r^\pm$; (ii) the shortest path from either endpoint $r^\pm$ to $v \in g_r$ has total weight equal to the distance from $r^\pm$ to $C'$. Finally, we insert into $A^\text{PATH}$ the set of routing edges $A^\text{Rte}$ from Section 8.3.2, with weights $w^\text{Rte}$. These weights are chosen so that the shortest path on $A^\text{Rte}$ from $i \in V$ to $j \in V$ has total weight equal to $\mathcal{D}(i, j)$.

**Proposition 8.12.** For any road network $R$, argument distributions $\mu^\natural$ and $\mu^\circ$ satisfying Assumptions 8.1 and 8.3, and $\epsilon > 0$, let $C_\epsilon$ denote the $\epsilon$-tessellation of $R$, let $N^\text{APPROX}$ denote the Wasserstein network generated by Section 8.5.1, with weights $w^\text{LOWER}$, and let $N^\text{PATH}$ denote the network generated by Section 8.6.2 with weights $w^\text{PATH}$. $(N^\text{APPROX}, w^\text{LOWER})$ and $(N^\text{PATH}, w^\text{PATH})$ are equivalent in the sense of Lemma 8.9.

The reasoning behind the proposition is the same as that of the construction. We omit the redundant formal proof.

Combining Proposition 8.12 and Lemma 8.9 shows that $\min_{f \in N^\text{PATH}} J(f; w^\text{PATH}) = \mathcal{W}$, and so proves its convergence to $\mathcal{W}(\mu^\natural, \mu^\circ)$ from below as $\epsilon \to 0^+$.

### 8.7 Analysis of Exact and Approximation Algorithms

In this section we analyze the complexity of construction of the three networks $N^\mathcal{W}$, $N^\text{APPROX}$, and $N^\text{PATH}$. In particular, we consider the way that the sizes of the instance graphs relate to (i) the size of the road network $R$ (both the size of its graphical representation and its physical size as determined by the lengths of roads); and (ii) the fine-ness $\epsilon$ of the input tessellation (in the case of approximation). Finally, we present a numerical study of graph sizes, approximation quality, and the runtime of a standard QP-based algorithm to compute each solution for the example network of Figure 8-3.

#### 8.7.1 Complexity

The remarkable feature of $N^\mathcal{W}$ is that it depends only on the size of the representation of $R$, and not on its physical size. $V^\mathcal{W}$ has size equal to $|V| + |R|$, and $A^\mathcal{W}$ has size
bounded by \(4|R|\); there are exactly two decision edges and as many as two routing edges per road \(r \in R\). The size of \(N^{\text{APPROX}}\), on the other hand, depends on the physical size of the network and on the approximation parameter \(\epsilon\). \(V^{\text{APPROX}}\) has size equal to \(2|C_e|\) or \(2 \sum_{r \in R} N_r\), which goes as \(\Theta(1/\epsilon)\). \(A^{\text{APPROX}}\) has size equal to \(|C_e|^2\), which has dominating complexity \(\Theta(1/\epsilon^2)\). Note that such growth of \(N^{\text{APPROX}}\) may be quite impractical to approximate the EMD with realistic road networks with hundreds or even thousands of miles of streets. \(N^{\text{PATH}}\) leverages the structure of the road network to reduce the space complexity of approximation to \(O(1/\epsilon)\). \(V^{\text{PATH}}\) has size equal to \(|V| + |C_e|\) and \(A^{\text{PATH}}\) has size bounded by \(2|R| + 2|C_e|\). Note that the size of \(N^{\text{PATH}}\) depends on both the physical size of the road network and the size of its representation.

### 8.7.2 Numerical Study

Figure 8-7(a) shows a plot of the number of vertices instantiated in \(N^{v}\), \(N^{\text{APPROX}}\), and \(N^{\text{PATH}}\), as a function of \(\epsilon\), for the EMD problem discussed in Section 8.4 (Figure 8-3). Figure 8-7(b) shows a plot of the number of edges instantiated. \(N^{v}\) exhibits a flat response to \(\epsilon\) in both plots, since it does not depend on the parameter. As expected, both approximation schemes exhibit the same rate of growth (\(\Omega(1/\epsilon)\)) in the number of vertices instantiated, while \(N^{\text{APPROX}}\) has a factor \(1/\epsilon\) greater growth in the rate of edges instantiated.

Figure 8-8(a) shows a plot of the quality of approximation of the methods in Sections 8.5.1 and 8.6.2, respectively, for values of the resolution parameter \(\epsilon\) as small as possible under space and runtime considerations (e.g., producing less than 100,000 graph objects, and running in minutes on an Intel i5 processor with 4 CPUs and 4GB of RAM). The dashed center line marks the solution obtained by the exact algorithm, i.e., optimization over the flow network in Figure 8-4. The plot shows convergence of the approximation bounds to the value predicted by \(N^{v}\).
Figure 8.8: Quality of the approximation and runtime of the algorithm, versus the fine-ness parameter $\epsilon$ of the $\epsilon$-tessellation of the roadmap in Figure 8.3, to estimate $\mathcal{W}$ between the distributions illustrated in the same figure. The flat line in 8.8(a) indicates performance achieved using $\mathcal{N}^\mathcal{W}$ (the proposed algorithm), which does not depend on $\epsilon$.

8.8 Limit Analysis for Path-based Approximation

$\mathcal{N}^{\text{PATH}}$ is sufficiently structured that it will allow us to calculate the limit of (8.19) as $\epsilon \to 0^+$. As argued in Section 8.5.2, that limit is equal to the EMD between the argument distributions. In this section we present a derivation of the limit, which produces the formulation of $\mathcal{N}^\mathcal{W}$ in Section 8.3.2.

Suppose we are trying to compute the EMD between $\mu^\mathcal{W}$ and $\mu^\mathcal{R}$ over a road network $\mathcal{R}$. Let $\mathcal{N}^\mathcal{W}$ denote the resulting $\mathcal{W}$ Wasserstein network, with edge costs $c^\mathcal{W}$; let $\mathcal{N}^{\text{PATH}}$ be the PATH network generated by some $\epsilon$-tessellation of $\mathcal{R}$, with weights $w^{\text{PATH}}$. Note that the routing edges $A^{\text{Rte}}$ are present in both networks, so the two networks differ only between the decision edges $A^{\text{Dec}}$ in $\mathcal{N}^\mathcal{W}$ and the road devices in $\mathcal{N}^{\text{PATH}}$.

8.8.1 Costs Associated with Road Devices

Let $f^*$ be a minimum-cost flow on $\mathcal{N}^{\text{PATH}}$, and let us consider the cost associated with the device $g_r$ of a non-transshipment road $r \in S \cup D$. As in Figure 8.6, let the vertices of $g_r$ be ordered $(u_r^1, u_r^2, \ldots, u_r^N)$ from $r^-$ to $r^+$. Suppose $r \in S$. Then from inspection of the device in Figure 8.6, we can denote the cost associated with $g_r$ by

$$J_r(f^*; w^{\text{PATH}}) = \sum_{k=1}^{N-2} \epsilon_r f^*(u_r^k, u_r^{k+1}) + \epsilon_r f^*(u_r^{k+1}, u_r^k).$$

(8.22)

Let us call all the edges of the form $(u_r^k, u_r^{k+1})$ the forward edges; in a similar fashion,
we call all the edges of the form \((u_r^k, u_r^{k-1})\) the backward edges; here, we are letting \(u_r^0\) and \(u_r^{N+1}\) denote symbolically the vertices \(r^-\) and \(r^+\) (respectively). Our ability to obtain a meaningful expression relies crucially on an important technical property of minimum-cost flows on PATH networks:

Note that between any adjacent vertices in \(g_r\), positive flow can be supported only either on the forward edge or the backward edge; otherwise, \(f^*\) would be non-minimal by existence of a cycle. We say that a vertex \(u_r^k\) "parts" device \(g_r\) if all forward flows (i.e., positive flows on forward edges) are on one side of \(u_r^k\) and all backward flows are on the opposite side.

Lemma 8.13 (Minimum-cost flows part road devices). Let \(f^*\) be a minimum-cost admissible flow on \(N^\text{PATH}\), generated by some \(\epsilon\)-tesselation of some road network \(\mathcal{R}\). Then every road device in \(N^\text{PATH}\) is parted by \(f^*\).

Proof. The proof is by contradiction: Assume that \(f^*\) is a minimum-cost admissible flow, but the device of some \(r \in S\) is not parted. (We give the proof only for \(r \in S\), but the proof for \(r \in D\) is symmetrical.) Note that because \(r \in S\), then \(b^\text{PATH}(u_r^k) \geq 0\) for \(k = 1, \ldots, N\). This implies that the backward flows are non-decreasing in magnitude from \(r^+\) to \(r^-\) and the forwards flows are non-decreasing from \(r^-\) to \(r^+\). (Otherwise, \(f^*\) would be either non-minimal, by existence of a positive-flow cycle, or else not admissible, by violation of a flow conservation constraint.) Since \(g_r\) is not parted by assumption, then the flow changes direction at least twice. Thus, there are indices \(k'\) and \(k''\), \(k' \leq k''\), such that \(f(u_r^{k'}, u_r^{k'+1}) > 0\) and \(f(u_r^{k''+1}, u_r^{k''}) > 0\). In that case, the monotonicity of forward and backward flows implies the existence of a positive-flow cycle somewhere between \(k'\) and \(k''\), drawing a contradiction against optimality of \(f^*\).

The parting of the road devices is quite powerful, because in combination with the flow conservation constraints (7.7), it allows us to express the whole device cost (8.22) in terms of the known supplies \(b^\text{PATH}\), and ultimately, the density function \(\varphi_r\).

Lemma 8.14 (Costs of Parted Devices). Let \(N^\text{PATH}\) be the PATH Wasserstein network for some \(\epsilon\)-tesselation of a road network \(\mathcal{R}\) with argument distributions \(\mu_r^*\) and \(\mu_r^\sharp\). Let \(r\) be some non-transshipment road and let \(f\) be any admissible flow on \(N^\text{PATH}\) which parts \(r\); let \(k_r\) denote the index of the part of \(g_r\). Then

\[
J_r(f; w^\text{PATH}) = o(1) + \int_{y=0}^{k_r \times \epsilon_r} \varphi_r(y) \ dy + \int_{y=k_r \times \epsilon_r}^{L_r} \varphi_r(y) \ [L_r - y] \ dy, \quad (8.23)
\]

\[
f(t\text{conn}_r) = \Phi(k_r \times \epsilon_r; \varphi_r) + o(1), \quad \text{and} \quad (8.24)
\]

\[
f(h\text{conn}_r) = \Phi(L_r - k_r \times \epsilon_r; \chi_r) + o(1). \quad (8.25)
\]

The proof of the lemma is fairly technical, and is provided in Appendix C.2.
Lemma 8.15 (Costs of Parted Devices, asymptotic). Let \( r \) be a non-transshipment road and let \( f \) be any admissible flow on \( \mathcal{N}^\text{PATH} \) which parts \( r \). Then
\[
J_r(f; \mathbf{w}^\text{PATH}) = q(f(tconn_r); \varphi_r) + q(f(hconn_r); \chi_r) + o(1).
\] (8.26)

Proof. It is easy to show that
\[
\int_0^y \varphi(y') \, dy' = q(\Phi(y; \varphi); \varphi).
\]

Thus, we can obtain the first term of (8.26) by combining the first integral of (8.23) with (8.24), and saving off any low order terms (recall our assumption that all \( q \) are Lipschitz). Similarly, we can obtain the second term of (8.26) by combining the second integral of (8.23) with (8.25); in that case, first, we put a change of variables \( y' = L_r - y \) and a substitution by \( \chi_r \).

8.8.2 Proving the Main Result

Lemma 8.15 provides the critical component of the proof of the main result of the chapter, i.e., Theorem 8.2.

Proof of Theorem 8.2. We begin by proving that \( \min_{f \in \mathcal{N}^\ast} J(f; \mathbf{c}^\ast) \leq \mathcal{W}(\mu^1, \mu^2) \). That proof is by showing that
\[
\min_{f \in \mathcal{N}^\ast} J(f; \mathbf{c}^\ast) = o(1) + \min_{f \in \mathcal{N}^\text{PATH}} J(f; \mathbf{w}^\text{PATH}),
\] (8.27)
where \( \mathcal{N}^\text{PATH} \) is the \( \epsilon \)-tesselation of \( \mathcal{R} \) for \( \epsilon > 0 \) arbitrarily small, so that the lemma holds in the limit as \( \epsilon \to 0^+ \). Let \( f^\ast \) be a minimum-cost admissible flow on \( \mathcal{N}^\text{PATH} \), and let \( f \) be the network flow on \( \mathcal{N}^\ast \) defined by
\[
f(tconn_r) := f^\ast(tconn_r)
\]
\[
f(hconn_r) := f^\ast(hconn_r) \quad \text{for all } r \in S \cup D, \text{ and}
\]
\[
f(a) := f^\ast(a) \quad \text{for all } a \in A^\text{Rte}.
\] (8.28-8.30)

It is a simple exercise to show that \( f \) is admissible, i.e., \( f \in \mathcal{N}^\ast \). Applying Lemma 8.15, we observe that for every road \( r \in R \), the difference between the cost of the road device \( g_r \) in \( \mathcal{N}^\text{PATH} \) and the combined cost of the decision edges \( tconn_r \) and \( hconn_r \) in \( \mathcal{N}^\ast \) is \( o(1) \). The flows and weights on \( A^\text{Rte} \) are identical in both networks, contributing no additional differences. Therefore, the total difference in cost between \( f \) and \( f^\ast \) is \( o(1) \). The minimum-cost flow on \( \mathcal{N}^\ast \) cannot have greater cost than achieved by \( f \), and so we obtain (8.27).

We prove the matching lower bound by another limiting expression
\[
\min_{f \in \mathcal{N}^\text{PATH}} J(f; \mathbf{w}^\text{PATH}) \leq o(1) + \min_{f \in \mathcal{N}^\ast} J(f; \mathbf{c}^\ast),
\] (8.31)
Let $f^*$ be a minimum-cost admissible flow on the flow network $\mathcal{N}^w$. We shall construct an admissible flow $f \in \mathcal{N}^{\text{PATH}}$. In particular, let $f$ be the flow satisfying again (8.28), (8.29), and (8.30), and parting every device $g_r$. Such $f$ can be generated, e.g., by traversing each $g_r$ from tail to head, starting with (8.28), and assigning the remaining flows in accordance with conservation constraints while avoiding the creation of flow cycles. The rest of the proof continues by symmetrical logic.

8.9 Conclusion

In this chapter we have defined the Earth Mover’s distance with respect to the ground metrics supplied by road networks, e.g., as defined in Chapter 6 (Section 6.2). We produced a formulation for the EMD for a general class of probability distributions which is explicit in the sense that it is amenable to efficient computational optimization techniques; it generalizes the formula of the EMD on the line, which has been one of few explicit results known previously. In the case that both distributions are piece-wise uniform, the EMD can be computed by quadratic programming.

Future Work: There are several directions is which this work may be extended. For example, the authors are quite certain that the basic formulation shall admit simple extensions for (i) the class of mixed distributions, i.e., distributions having an absolutely continuous part and an atomic part; (ii) non-symmetrical ground metrics resulting from the treatment of “one-way” streets. It should also be straightforward to obtain a generalization of the formulation for definitions of the EMD (e.g., in [100]) which allow input measures to have unequal total mass. Another possible extension of this work would be to obtain faster algorithms for road networks with special structure. (For example, it should be possible to produce an algorithm in the style of [77, Sec. 5.3] for road networks that can be represented by tree graphs.)

In addition to these particular extensions, we hope that our formal treatment of road networks and the analytical techniques introduced in this chapter may facilitate bringing the power of computational statistics methodologies to bear on basic research questions framed in the ubiquitous road network setting.
Chapter 9

Conclusion

Part I. A primary contribution of Part I of the thesis has been a stability theory for the 1-DPDP, with the novel property of having a hidden intrinsic service time component; the hidden Earth Mover's distance term has indeed been wondered at in the past, but not specifically characterized. The stability theory and the techniques developed to derive it should be regarded as an enabling theory for future study, while the main focus of queueing analysis for DVRPs is to derive fundamental performance limitations, and to devise provably good routing policies which can be implemented using real-time, scalable algorithms.

One of the most obvious new challenges and open questions arising about the 1-DPDP is to derive general, policy-independent lower bounds governing important statistics like average steady-state queue length or system time. (The sentiment in the community is indeed that the existing bounds are more likely to yield from below.) This is an important research direction, because novel techniques are required (not based on the "trivial" statistics of nearest-neighbors) for improving system time lower bounds to match the guarantees given by "good" policies. (A closely related research goal is to derive tighter probabilistic convergence results for bipartite matching and SCP functionals with general demand distributions, i.e., \( f_Y \neq f_X \); our results in Chapter 5 are tight only in the term linear in \( n \); a lower bound for the next-order component(s) remains an open question.)

A considerable strength of the sizable literature on other DVRPs, like the DTRP and its variants, is a strong focus on finding scalable policies with strong performance guarantees. In the multi-robot setting, system designers prefer algorithms that are not only tractable, but also distributed or decentralized, with low communication cost both in terms of infrastructure and/or messaging. An important, non-trivial research direction is to ascertain the degree to which scheduling and routing for shared vehicles is distributable, for the sake of scale.

Other considerations of import are sensitivity to "information constraints": For example, one might consider the case that the destination of a particular demand is revealed only at the time that a vehicle reaches the corresponding origin point, or even that vehicles must search the network to discover the physical manifestation of demands at their origins.

Part II. In the second part of the thesis, we sought to mitigate a particularly
vexing limitation of previous analytical techniques in practice: that they ignore the extremely common “street-rules” of ground-based environments. First, we presented simple extensions of most of the analytical findings of the first part of the thesis. Then, we discovered that several algorithmic improvements are possible on road networks in practice. We gave a new efficient algorithm for the Bipartite Matching problem for points on a roadmap, with the potential to greatly decrease the computational burden of centralized planning and vehicle routing. We also derived a new explicit formulation of the Earth Mover’s distance on roadmaps, making numerical analysis significantly easier on road networks than in the plane.

An interesting direction of research, with greater importance on road networks than in the plane, would be to consider the effects of traffic congestion on system performance in the multi-vehicle setting.
Appendix A

Technical Proofs for Chapter 3

Proof of Lemma 3.9. Although it is intuitive, we begin by proving boundedness. We will prove the limit afterward. Without loss of generality, we assume $e$ is non-decreasing, since we can always find a continuous $o(y^γ)$ function to bound it which is. Letting $f(y) := e(y)/y^γ$, we can write (3.20) as

$$y(i + 1) = ay(i) + \left(b + f(y(i))\right)y(i)^γ.$$ 

Note that $f(z) = o(1)$, i.e., $\lim_{z\to\infty} f(z) = 0$. Since $f$ is also continuous, we can bound the second term by $By^γ$ for some $B$ sufficiently large.

Now, let $\varepsilon$ denote some small constant, $\varepsilon > 0$, which shall be determined later. From Young's inequality

$$t = \frac{t(p\varepsilon)^α}{(p\varepsilon)^α} \leq \left(t(p\varepsilon)^α\right)^\frac{1}{p} + \left(\frac{1}{(p\varepsilon)^α}\right)^\frac{1}{q},$$

for all $t \in \mathbb{R}_{>0}$, for any choice of $p, q \in \mathbb{R}_{>0}$, $1/p + 1/q = 1$ and $α \in \mathbb{R}_{>0}$. By letting $t = y^γ$, $p = 1/γ$, $α = γ$, and $q = 1/(1-γ)$, we obtain:

$$y^γ \leq \varepsilon y + (1-γ)\left(γ/ε\right)^{γ/(1-γ)}.$$

Let us denote by $C(ε)$ the constant $(1-γ)(γ/ε)^{γ/(1-γ)}$. Next, we define System-Z as

$$z(i + 1) = az(i) + B\left[ε z(i) + C(ε)\right].$$

starting from $z(1) = y(1)$. It is immediate to show that System-Z bounds System-Y, i.e.,

$$y(i) \leq z(i), \quad \text{for all } i \geq 1.$$  

(A.2)

The proof now proceeds by showing that the trajectories of System-Z are bounded; this fact, together with equation (A.2), implies that also trajectories $y(i)$ are bounded.

System-Z is a discrete-time linear system and can be rewritten in compact form as

$$z(i + 1) = \left(a + \varepsilon B\right)z(i) + BC(ε).$$

159
Since $a < 1$, then there exists a sufficiently small $\varepsilon > 0$ such that $a + \varepsilon B < 1$. Accordingly, having selected a sufficiently small $\varepsilon$, the solution $i \mapsto z(i) \in \mathbb{R}_{\geq 0}$ of System-Z converges exponentially fast to the unique equilibrium point

$$z^*(\varepsilon) = (1 - a - \varepsilon B)^{-1} BC(\varepsilon). \quad (A.3)$$

Combining equation (A.2) with the previous statement, we see that the solution $i \mapsto y(i)$ is bounded.

Now, we turn our attention to the limit of $y = \limsup_{i \to +\infty} y(i)$ as the coefficient $a \to 1^-$. Taking the limsup of the left- and right-hand sides of equation (3.20) we obtain

$$y \leq a y + b y^{\gamma} + e(y). \quad (A.4)$$

The limit $\limsup_{i \to +\infty} y(i)^\gamma + e(y(i)) = y^{\gamma} + e(y)$, because both mappings are continuous and non-decreasing on $\mathbb{R}_{\geq 0}$, so the limsup can be pulled inside.) Rearranging we obtain

$$y(1 - a)^{1/(1 - \gamma)} \leq (b + e(y)/y^{\gamma})^{1/(1 - \gamma)}. \quad (A.5)$$

Now, we take the lim sup on both sides as $a \to 1^-$. Recalling that $e(y)/y^{\gamma} = f(y) = o(1)$, then we obtain the lemma as $a \to 1^-$ as long as if $y \to +\infty$. On the other hand, if $y$ remains finitely bounded, then the lemma holds trivially.

Proof of Lemma 3.10. The present proof follows closely the logic of the previous one, with a twist inspired by the proof in [130, Sec. 4.4]. Let us start from the point that $y = \limsup_{i \to +\infty} y(i) < +\infty$. (It can be proved by the logic above, since $\sqrt{y \log y} = o(y^{\gamma})$ for any $\gamma > \frac{1}{2}$.) Since $x \mapsto x \log x$ is also continuous, and strictly, monotonically increasing, we have again that $y = a y + b \sqrt{y \log y} + e(y)$. Rearranging, we obtain

$$y(1 - a) = [b + o(1)] \sqrt{y \log y}. \quad (A.6)$$

Squaring on both sides, and rearranging again, we obtain

$$y = \frac{[b + o(1)]^2 \log y}{(1 - a)^2}. \quad (A.5)$$

Letting $\gamma := b/(1 - a)$, and ignoring the $o(1)$ term (it is straightforward but cumbersome to include it in the analysis, and it does not change the limiting behavior), we can express (A.5) as $y = \gamma^2 \log y$. Thus, $y$ is the unique positive root of the function $f(y) = y - \gamma^2 \log y$. Let us factor $y$ as

$$y = k(\gamma) \gamma^2 \log \gamma, \quad (A.6)$$

and consider the problem of solving for $k(\gamma)$. Substituting this factorization in $f(y)$, we obtain

$$0 = f(k) = k \gamma^2 \log \gamma - \gamma^2 \log \left( k \gamma^2 \log \gamma \right) = \gamma^2 \left\{ [k - 2] \log \gamma - \log k - \log \log \gamma \right\}. \quad (A.6)$$

Because $\log \gamma$ is the dominant term in the right-hand factor of the last expression (as $a \to 1^- (\gamma \to +\infty)$), we have that $\lim_{a \to 1^-} k = 2$. Multiplying both sides of (A.6) by
\[(1 - a)^2 / \log \left( \frac{1}{1-a} \right), \text{ and taking the limit as } a \to 1^-, \text{ we obtain} \]

\[
\lim_{a \to 1^-} y \frac{(1 - a)^2}{\log \left( \frac{1}{1-a} \right)} = \lim_{a \to +\infty} k \lim_{a \to +\infty} \gamma^2 (1 - a)^2 \lim_{a \to +\infty} \frac{\log \gamma}{\log \left( \frac{1}{1-a} \right)}.
\]

The component limits are 2, \(b^2\), and 1, respectively, obtaining the lemma. \(\square\)
Appendix B

Technical Proofs for Chapter 5

B.1 Distribution of Permutations Induced by Optimal Bipartite Matching

In this section we provide the rigorous proof of Lemma 5.5 from Section 5.4.3, showing that every permutation is equally likely to be produced by an optimal bipartite matching algorithm \( \mathcal{M} \). Let \( \mathcal{X}_n = \{x_1, \ldots, x_n\} \) and \( \mathcal{Y}_n = \{y_1, \ldots, y_n\} \) be two sets of points in \( \mathcal{W} \subset \mathbb{R}^d \). Consider \( b = \text{concat}(x_1, y_1, \ldots, x_n, y_n) \), a column vector formed by vertical concatenation of \( x_1, y_1, \ldots, x_n, y_n \). Note that the set \( \mathcal{W}^{2n} \) of such vectors is a full-dimensional subset of \( \mathbb{R}^{d(2n)} \). Let \( \Pi^* : \mathcal{W}^{2n} \to 2^{\Pi_n} \) denote the optimal permutation map that maps a batch \( b \in \mathcal{W}^{2n} \) into the set of permutations that correspond to optimal bipartite matchings (recall that there might be multiple optimal bipartite matchings). Let us denote the set of batches that lead to non-unique optimal bipartite matchings as:

\[
Z := \left\{ b \in \mathcal{W}^{2n} \left| |\Pi^*(b)| > 1 \right. \right\},
\]

where \( |\Pi^*(b)| \) is the cardinality of set \( \Pi^*(b) \).

In Lemma 5.5, \( \mathcal{M} \) may be any algorithm that computes an optimal bipartite matching. According to our definitions, the behavior of such an algorithm can be described as follows: given a batch \( s \in \mathcal{W}^{2n} \) it computes

\[
\mathcal{M}(b) = \begin{cases} 
\text{unique } \sigma \in \Pi^*(b) & \text{if } b \in \mathcal{W}^{2n} \setminus Z, \\
\text{some } \sigma \in \Pi^*(b) & \text{otherwise.}
\end{cases}
\]

Thus, the behavior of a bipartite algorithm on the set \( Z \) can vary; on the other hand, we now show that set \( Z \) has Lebesgue measure zero so that the behavior of an algorithm on this set is immaterial for our analysis.

**Lemma B.1 (Measure of multiple solutions).** The set \( Z \) has Lebesgue measure equal to zero.

**Proof.** The strategy of the proof is to show that \( Z \) is the subset of a set that has zero Lebesgue measure.
For \( \sigma', \sigma'' \in \Pi_n, \sigma' \neq \sigma'' \), let us define the sets:
\[
\mathcal{H}_{\sigma', \sigma''} := \left\{ b \in W^{2n} \bigg| \sum_{i=1}^{n} \| x_{\sigma'(i)} - y_i \| = \sum_{i=1}^{n} \| x_{\sigma''(i)} - y_i \| \right\};
\]
let us also define the union of such sets:
\[
\mathcal{H} := \bigcup_{\sigma', \sigma'' \in \Pi_n, \sigma' \neq \sigma''} \mathcal{H}_{\sigma', \sigma''}.
\]

The equality constraint in the definition of \( \mathcal{H}_{\sigma', \sigma''} \) implies that \( \mathcal{H}_{\sigma', \sigma''} \subseteq \mathbb{R}^{d(2n)-1} \), which has zero Lebesgue measure in \( \mathbb{R}^{d(2n)} \). Hence, the Lebesgue measure of \( \mathcal{H}_{\sigma', \sigma''} \) is zero. Since \( \mathcal{H} \) is the union of finitely many sets of measure zero, it has zero Lebesgue measure as well.

We conclude the proof by showing that \( Z \subseteq \mathcal{H} \). Indeed, if \( b \in Z \), then there must exist two permutations \( \sigma' \neq \sigma'' \) such that \( \sum_{i=1}^{n} \| x_{\sigma'(i)} - y_i \| = \min_\sigma \sum_{i=1}^{n} \| x_{\sigma(i)} - y_i \| \) and \( \sum_{i=1}^{n} \| x_{\sigma''(i)} - y_i \| = \min_\sigma \sum_{i=1}^{n} \| x_{\sigma(i)} - y_i \| \), i.e., there must exist two permutations \( \sigma' \neq \sigma'' \) such that
\[
\sum_{i=1}^{n} \| x_{\sigma'(i)} - y_i \| = \sum_{i=1}^{n} \| x_{\sigma''(i)} - y_i \|;
\]
which implies that \( b \in \mathcal{H} \). Hence, \( Z \subseteq \mathcal{H} \) and, therefore, it has zero Lebesgue measure.

Now we present the proof of Lemma 5.5, which gives the probability that \( \mathcal{M} \) produces as a result the permutation \( \sigma \) given point sets \( X_n \) and \( Y_n \); we call such probability \( \mathbb{P}[\sigma] \).

Proof of Lemma 5.5. We start by observing that it is enough to consider a restricted sample space, namely \( W^{2n} \setminus Z \). Because of our continuity assumptions on probability distributions, Lemma B.1 implies \( \mathbb{P}[b \in Z] = 0 \). Thus, by the total probability law,
\[
\mathbb{P}[\sigma] = \mathbb{P}[\mathcal{M}(b) = \sigma | b \in W^{2n} \setminus Z]. \tag{B.1}
\]
For each permutation \( \sigma \in \Pi_n \), let us define the set
\[
S_\sigma := \left\{ b \in W^{2n} \setminus Z | \mathcal{M}(b) = \sigma \right\}.
\]
Clearly, \( \mathbb{P}[\sigma] = \mathbb{P}[b \in S_\sigma] \). Collectively, sets \( S_\sigma \) form a partition of \( W^{2n} \setminus Z \).

Now, for each permutation \( \sigma \in \Pi_n \), let us define the reordering function \( g_\sigma : W^{2n} \to W^{2n} \) as the function that maps a batch \( b = \text{concat}(x_1, y_1, \ldots, x_n, y_n) \) into a batch \( b' = \text{concat}(x_{\sigma(1)}, y_1, \ldots, x_{\sigma(n)}, y_n) \). Alternatively, let \( F_j \in \mathbb{R}^{d \times 2nd} \) be a block row matrix of \( 2n \times d \) square blocks whose elements are equal to zero except the \((2j-1)\)th block that is identity; let \( F_j \in \mathbb{R}^{d \times 2nd} \) be such a block matrix, but whose elements are all zero except the \(2j\)th block that is identity. Then in matrix form the reordering function can be written as \( g_\sigma(b) = P_\sigma b \), where \( P_\sigma \) is the \( 2nd \times 2nd \) matrix.
defined by

\[ P_\sigma := \left[ E_{\sigma(1)}^T F_1^T E_{\sigma(2)}^T F_2^T \ldots E_{\sigma(n)}^T F_n^T \right]^T. \]

Note that \(|\det(P_\sigma)| = 1\) for all permutations \(\sigma\); also, Property Prop. 3 of Section 2.2.1 implies \(P_{\sigma^{-1}} = P_\sigma^{-1}\).

For every permutations \(\sigma \in \Pi_n\), \(g_\sigma\) has the important property that \(\{b \in S_\sigma\} \equiv \{g_\sigma(b) \in S_{\sigma_1}\}\), recalling that \(\sigma_1\) denotes the identity permutation (i.e., \(\sigma(i) = i\) for \(i = 1, \ldots, n\)). The property is quite easy to prove:

\[ b \in S_\sigma \implies g_\sigma(b) \in S_{\sigma_1}: \] Since \(b \in S_\sigma\), we have that \(\sum_{i=1}^n \|x_{\sigma(i)} - y_i\| = \min_{\sigma' \in \Pi_n} \sum_{i=1}^n \|x_{\sigma'(i)} - y_i\|\); moreover \(\sigma\) is the unique minimizer. \(g_\sigma(b) = : b'\) has the form \(\text{concat}(x_{\sigma(1)}, y_1, \ldots, x_{\sigma(n)}, y_n)\). Indeed, \(\sigma_1\) is an optimal matching of \(b'\) (by inspection), i.e., \(\sigma_1 \in \Pi^*(b')\). Suppose, however, there is another optimal matching \(\sigma' \neq \sigma_1\) such that \(\sigma' \in \Pi^*(b')\). Then \(\sigma'\sigma\) is an optimal matching of \(b\); however, this is a contradiction, because \(\sigma' \neq \sigma_1 \implies \sigma'\sigma \neq \sigma\), which is the unique minimizer for \(b\).

\[ g_\sigma(b) \in S_{\sigma_1} \implies b \in S_\sigma: \] Suppose \(g_\sigma(b) = : b'\) has the form \(\text{concat}(x'_1, y'_1, \ldots, x'_n, y'_n)\). Since \(b' \in S_{\sigma_1}\), we have that \(\sum_{i=1}^n \|x'_{\sigma_1(i)} - y'_i\| = \min_{\sigma' \in \Pi_n} \sum_{i=1}^n \|x'_{\sigma'(i)} - y'_i\|\) (again, uniquely). \(g_\sigma\) is an invertible mapping with \(g_\sigma^{-1} = g_{\sigma^{-1}}\), e.g., since \(P_{\sigma^{-1}}\) exists and is equal to \(P_{\sigma^{-1}}\). Then \(b = g_{\sigma^{-1}}(b')\) has the form \(\text{concat}(x'_{\sigma^{-1}(1)}, y'_1, \ldots, x'_{\sigma^{-1}(n)}, y'_n)\). By inspection, a permutation \(\sigma'\) that satisfies \(\sigma'\sigma^{-1} = \sigma_1\) must be a solution in \(\Pi^*(b)\). Since \(\sigma_1\) is the unique optimal matching for \(b'\), the condition is also necessary. Thus, \(\sigma\) is the unique optimal matching for \(b\).

We are ready to evaluate the probabilities of permutations as follows: One can write \(\mathbb{P}[b \in A] = \int_A \varphi(b) \, db\), for any region \(A\), where \(\varphi(b)\) denotes the product \(\prod_{i=1}^n \varphi_X(x_i)\varphi_Y(y_i)\). For any permutation \(\sigma\) we have \(\mathbb{P}[\sigma] = \mathbb{P}[S_\sigma] = \int_{S_\sigma} \varphi(b) \, db\). We use variable substitution \(b' = g_\sigma(b) = P_\sigma b\) and the property \(g_\sigma(S_\sigma) = S_{\sigma_1}\), and we apply the rule of integration by substitution to obtain:

\[
\int_{S_\sigma} \varphi(b) \, db = \int_{S_{\sigma_1}} \varphi \left( P_{\sigma^{-1}} b' \right) \det P_{\sigma^{-1}}^{-1} \, db'.
\]

Observing that

\[ \varphi(P_{\sigma^{-1}} b) = \varphi(P_{\sigma^{-1}} b) = \prod_{i=1}^n \varphi_X(x_{\sigma^{-1}(i)})\varphi_Y(y_i), \]

and that

\[ \prod_{i=1}^n \varphi_X(x_{\sigma^{-1}(i)})\varphi_Y(y_i) = \prod_{i=1}^n \varphi_X(x_i)\varphi_Y(y_i) = \varphi(b), \]

we obtain

\[
\int_{S_{\sigma_1}} \varphi(P_{\sigma^{-1}} b) \, db = \int_{S_{\sigma_1}} \varphi(b) \, db = \mathbb{P}[S_{\sigma_1}] = \mathbb{P}[\sigma_1].
\]
Combining these results, we conclude $P[\sigma] = P[\sigma_1]$ for all $\sigma \in \Pi_n$, obtaining the lemma. \hfill \Box

## B.2 Cost of Bipartite Matching, Lower Bounds

**Proof of Lemma 5.7.** Fix $n$, and let $\sigma$ denote the permutation of the optimal bipartite matching between the first $n$ elements of $\{X_k\}$ and $\{Y_k\}$. For a particular partition $C$, we define random variables $\hat{t}_{ij} := \left| \{k \leq n : Y_k \in C^i, X_{\sigma(k)} \in C^j\} \right| /n$ for every pair $(C^i, C^j)$ of cells; that is, $\hat{t}_{ij}$ denotes the fraction of matches under $\sigma$ whose $Y$-endpoints are in $C^i$ and whose $X$-endpoints are in $C^j$. Let $\hat{\mathcal{F}}_n$ be the set of matrices with entries $\{\hat{t}_{ij} \geq 0\}_{i,j=1,...,|C|}$ such that $\sum_i \hat{t}_{ij} = |X_n \cap C^j| /n$ for all $C^j \in C$ and $\sum_j \hat{t}_{ij} = |Y_n \cap C^i| /n$ for all $C^i \in C$. Note that $\{\hat{t}_{ij}\}$ and the set $\hat{\mathcal{F}}_n$ are both random, but that $\{\hat{t}_{ij}\}$ is an element of $\hat{\mathcal{F}}_n$, surely. Thus, the term $n^{-1}M^*(n)$ is bounded below by

$$n^{-1}M^*(n) = n^{-1} \sum_{k=1}^n \|X_{\sigma(k)} - Y_k\| \geq \sum_{ij} \hat{t}_{ij} \min_{y \in C^i, x \in C^j} \|x - y\| \geq \min_{\hat{\mathcal{F}}_n} \sum_{ij} \hat{t}_{ij} \min_{y \in C^i, x \in C^j} \|x - y\|.$$ 

The right-hand side of the bound is still a random variable, however, the key observation is that $\lim_{n \to \infty} \left| \{X_n \cap C^j\} /n \right| = f_X(C^j)$, and $\lim_{n \to \infty} \left| \{Y_n \cap C^i\} /n \right| = f_Y(C^i)$, almost surely. Thus, applying standard sensitivity analysis (see Chapter 5 of [23]), one can show that the final expression converges almost surely to $\mathcal{W}_C$ (of Problem 5.1) as $n \to +\infty$.

Specifically, for any finite partition $C$ and $\epsilon > 0$, consider the sequence of events $E_n$

$$E_n = \left\{ (X_n, Y_n) : \left| -\mathcal{W}_C + \min_{\hat{\mathcal{F}}_n} \sum_{ij} \hat{t}_{ij} \min_{y \in C^i, x \in C^j} \|x - y\| \right| > \epsilon \right\}.$$

Consider Problem 5.1: let $\lambda^Y_i$ for all $i$ be the dual variables associated with the constraints $\sum_j t_{ij} = f_X(C^i)$; let $\lambda^X_j$ for all $j$ be the dual variables associated with the constraints $\sum_i t_{ij} = f_Y(C^j)$. (These are all finite, deterministic constants.) Now, let us define random variables $\Delta^Y_i(n) = |Y_n \cap C^i| /n - f_Y(C^i)$ for all $i$, and $\Delta^X_j(n) = |X_n \cap C^j| /n - f_X(C^j)$ for all $j$. Through sensitivity analysis we obtain

$$-\mathcal{W}_C + \min_{\hat{\mathcal{F}}_n} \sum_{ij} \hat{t}_{ij} \min_{y \in C^i, x \in C^j} \|x - y\| = -\sum_i \lambda^Y_i \Delta^Y_i(n) + o(\Delta^Y_i(n)) + \sum_j \lambda^X_j \Delta^X_j(n) + o(\Delta^X_j(n))$$
and so

\[ \begin{align*}
-\mathcal{W}_C + \min_{j_n} \sum_{i,j} t_{ij} \min_{y \in C_i, x \in C_j} \|x - y\| & \leq \sum_i |\lambda_i^Y| |\Delta_i^Y(n)| + o(\Delta_i^Y(n)) + \sum_j |\lambda_j^X| |\Delta_j^X(n)| + o(\Delta_j^X(n)).
\end{align*} \]

(B.2)

Thus, we can choose \( \epsilon_i^Y > 0 \) for all \( i \) and \( \epsilon_j^X > 0 \) for all \( j \) so that the right-hand side of (B.2) is less than or equal to \( \epsilon \) as long as \( |\Delta_i^Y| \leq \epsilon_i^Y \) for all \( i \) and \( |\Delta_j^X| \leq \epsilon_j^X \) for all \( j \). Let us now define the event functions \( E_i^Y(n) \) for all \( i \) and \( E_i^X(n) \) for all \( j \), where

\[ E_i^Y(n) = \{(X_n, Y_n) : |\Delta_i^Y| > \epsilon_i^Y\} \quad \text{and} \quad E_i^X(n) = \{(X_n, Y_n) : |\Delta_i^X| > \epsilon_j^X\}. \]

Note that the Strong Law of Large Numbers gives \( P[\limsup_{n \to \infty} E_i^Y(n)] = 0 \) for all \( i \), and \( P[\limsup_{n \to \infty} E_j^X(n)] = 0 \) for all \( j \). Observing that

\[ P[\limsup_{n \to \infty} E_n] \leq \sum_i P[\limsup_{n \to \infty} E_i^Y(n)] + \sum_j P[\limsup_{n \to \infty} E_j^X(n)], \]

we obtain the claim. \( \square \)

**Proof of Lemma 5.8.** First, we show that \( \mathcal{W}_C(f_Y, f_X) \leq \mathcal{W}(f_Y, f_X) \) for any finite partition \( C \). Let us define a distance approximation function

\[ \mathcal{D}_C(x_1, x_2) = \sum_{C, C' \in C} \mathbb{I}\{x_1 \in C, x_2 \in C'\} \min_{x \in C, x' \in C'} \|x' - x\|. \]

Note that \( \mathcal{D}_C \) is everywhere a lower bound for the Euclidean distance. Let us fix \( \gamma^* \), some solution arbitrarily close to the infimum of (5.2), say with difference \( \delta > 0 \). Noting that the matrix \( \{t_{ij} = \int_{C_i \times C_j} d\gamma^*(x_1, x_2)\} \) satisfies the constraints of Problem 5.1, we can write

\[ \mathcal{W}(f_Y, f_X) + \delta = \int_{x_1, x_2 \in \Omega} \|x_2 - x_1\| \ d\gamma^*(x_1, x_2) \geq \int_{x_1, x_2 \in \Omega} \mathcal{D}_C(x_1, x_2) \ d\gamma^*(x_1, x_2) = \sum_{ij} \min_{x \in C_i, y \in C_j} \|y - x\| \int_{x_1 \in C_i} \int_{x_2 \in C_j} d\gamma^*(x_1, x_2) \geq \mathcal{W}_C(f_Y, f_X). \]

Since this inequality holds for all \( C \) and \( \delta \) arbitrarily small, we conclude the first part.

Next, we show that for any \( \delta > 0 \), there exists a partition \( C \) so that \( \mathcal{W}(f_Y, f_X) \leq \mathcal{W}_C(f_Y, f_X) + \delta \). For example, we may choose a grid-based partition \( C \), with resolution \( r \) sufficiently fine so that \( \|x_2 - x_1\| < \mathcal{D}_C(x_1, x_2) + \delta \) everywhere. Let \( \{t_{ij}^*\} \) denote the
optimal matrix solution to Problem 5.1. There are generally many joint distributions $\gamma \in \Gamma(f_Y, f_X)$ which satisfy $\int_{x_1 \in C_1, x_2 \in C_2} d\gamma(x_1, x_2) = t_{ij}^*$ for all $i$ and $j$. For example, one may choose the unique $\gamma$ such that

$$\gamma(A \times B) = \sum_{ij} t_{ij}^* \frac{f_Y(A \cap C^i)}{f_Y(C^i)} \frac{f_X(B \cap C^j)}{f_X(C^j)}$$

for all measurable $A$ and $B$. Then we have

$$\mathcal{W}_C(f_Y, f_X) = \int \mathcal{Q}_C(x_1, x_2) \, d\gamma^*(x_1, x_2)$$

$$\geq \int \|x_2 - x_1\| \, d\gamma^*(x_1, x_2) - \delta \geq \mathcal{W}(f_Y, f_X) - \delta. \quad \square$$

B.3 Cost of Bipartite Matching, Upper Bounds

Proof of Lemma 5.11. By assumption, the points $Y_1, \ldots, Y_n$ are i.i.d. and independent of $X_1, \ldots, X_n$. Thus, under the explicitly point-wise independent sampling of Algorithm 2, the $n$ joint variables $\{(Y_k, J_k, X'_k)\}_{k=1}^n$ are i.i.d., and independent of $X_1, \ldots, X_n$. This suffices to prove (i) and the mutual independence of (ii).

Thus, we are left only to prove the distribution of $X'$. By inspection of Algorithm 2, we can write

$$f_{X'}(\cdot) = \sum_j f_X(\cdot; X \in C^j) \mathbb{P}[J = j].$$

We can show that $\mathbb{P}[J = j] = \mathbb{P}[X \in C^j]$ for all $j$ as follows:

$$\mathbb{P}[J = j] = \int \mathbb{P}[J = j \mid Y = y] \varphi_Y(y) \, dy$$

$$= \int \sum_i t_{ij} \frac{f_Y(C^i)}{f_Y(C^j)} \{y \in C^i\} \varphi_Y(y) \, dy$$

$$= \sum_i \frac{t_{ij}}{f_Y(C^i)} \int_{C^i} \varphi_Y(y) \, dy = \sum_i t_{ij} = \mathbb{P}[X \in C^j]. \quad \square$$

Thus, the result obtains from the law of total probability.

Proof of Lemma 5.12. We can write the expectation as

$$\mathbb{E}_T \|X'(Y) - Y\| = \int \varphi_Y(y) \mathbb{E}_T \|X'(y) - y\| \, dy.$$

and expand the expectation in the argument as

$$\mathbb{E}_T \|X'(y) - y\| = \sum_j \mathbb{E}_T \{\|X'(y) - y\| \mid J = j\} \mathbb{P}[J = j \mid Y = y].$$
A useful inequality for this proof is that
\[ \mathbb{E}_T \{ \|X'(y) - y\| \mid J = j \} = \int \|x - y\| \varphi_X(x \mid X \in C^j) \, dy \leq \max_{x \in C^j} \|x - y\|. \quad (B.3) \]

Thus, we can write
\[ \mathbb{E}_T \|X'(Y) - Y\| \leq \int \varphi_Y(y) \sum_j \max_{x \in C^j} \|x - y\| \sum_i \frac{t_{ij}}{f_Y(C_i)} \mathbb{I}\{y \in C^i\} \, dy \]
\[ = \sum_{ij} \frac{t_{ij}}{f_Y(C_i)} \int \varphi_Y(y) \max_{x \in C^j} \|x - y\| \mathbb{I}\{y \in C^i\} \, dy \]
\[ \leq \sum_{ij} \frac{t_{ij}}{f_Y(C_i)} \max_{y \in C^i, x \in C^j} \|x - y\| \int_{C^i} \varphi_Y'(y) \, dy, \]
\[ \square \]
where to obtain the last inequality, we bound the norm term by its maximum over \( C^i \), to bring it outside of the integral. Cancelling terms appropriately obtains the lemma.

**Proof of Lemma 5.13.** Algorithm 3 uses the grid-based partition \( C_r \) and an optimal solution \( T^* \) of Problem 5.2 as inputs to Algorithm 2. Note that for any feasible transportation matrix \( T \in \mathcal{S}_C \) we can write
\[ \mathbb{E} \|X'(Y) - Y\| \leq \sum_{ij} t_{ij}^* \max_{y \in C^i, x \in C^j} \|x - y\| \leq \sum_{ij} t_{ij} \max_{y \in C^i, x \in C^j} \|x - y\|. \]
The first inequality is by Lemma 5.12, and the second is the definition of optimality. In particular, let \( T \) be an optimal solution to Problem 5.1 with respect to the same partition \( C_r \); thus, \( \sum_{ij} t_{ij} \min_{y \in C^i, x \in C^j} \|x - y\| = \mathcal{W}_C(f_Y, f_X) \leq \mathcal{W}(f_Y, f_X) \) (see the proof of Lemma 5.8 in Appendix B.2). Combining the results above, and letting \( C^j - C^i \equiv \{x - y : y \in C^i, x \in C^j\} \), we obtain
\[ \mathbb{E} \|X'(Y) - Y\| - \mathcal{W}(f_Y, f_X) \leq \sum_{ij} t_{ij} \left[ \max_{\Delta \in C^j - C^i} \|\Delta\| - \min_{\Delta \in C^j - C^i} \|\Delta\| \right] \]
\[ \leq (2 \text{diam } \mathcal{W} \sqrt{d/r}) \sum_{ij} t_{ij} = 2 \text{diam } \mathcal{W} \sqrt{d/r}. \quad \square \]
Appendix C

Technical Proofs for Chapter 8

C.1 Correctness of the General Purpose Approximation

Before proving the two propositions, we must introduce a relation between the set of couplings $\Gamma(\mu^2, \mu^b)$ and the network flow constraints on $\mathcal{N}^{\text{APPROX}}$.

**Lemma C.1 (Coupling-induced network flow).** Let $\mu^2$ and $\mu^b$ be two measures over a domain $\Omega$. Let $C$ be a partition of $\Omega$ into cells, and let $\mathcal{N}^{\text{APPROX}}$ be the approximation network derived from $\mu^2$, $\mu^b$, and $C$. Let $\gamma$ be a coupling of measures $\mu^2$ and $\mu^b$, $\gamma \in \Gamma(\mu^2, \mu^b)$. Let $f : V^2 \times V^b \to \mathbb{R}$ be the mapping where

$$ f(u, u') = \gamma(C \times C') \quad \text{for each } (u, C) \in M^2 \text{ and } (u', C') \in M^b. \quad \text{(C.1)} $$

Then $f$ is admissible, i.e., $f \in \mathcal{N}^{\text{APPROX}}$.

**Proof.** To prove $f$ of C.1 is admissible, one must show that (7.7) holds. On the bipartite network $\mathcal{N}^{\text{APPROX}}$, (7.7) holds if $\sum_{(u', C') \in M^b} f(u, u') = \mu^b(C')$ for all $(u, C) \in M^2$ and $\sum_{(u, C) \in M^2} f(u, u') = \mu^2(C)$ for all $(u', C') \in M^b$. Recalling that $\gamma \in \Gamma(\mu^2, \mu^b)$, these conditions can be easily verified. \hfill \Box

**Proposition C.2.** For any admissible flow $f \in \mathcal{N}^{\text{APPROX}}$, there exists at least one coupling $\gamma \in \Gamma(\mu^2, \mu^b)$ satisfying (C.1). (In general, there are many.)

**Proof.** The proof is by an example construction. Given $f \in \mathcal{N}^{\text{APPROX}}$, let $\gamma$ be the unique measure satisfying

$$ \gamma(A \times B) = \sum_{(u, C) \in M^2, (u', C') \in M^b} f(u, u') \frac{\mu^2(A \cap C)}{\mu^2(C)} \frac{\mu^b(B \cap C')}{\mu^b(C')} $$

for all $A, B \in \mathcal{F}$ (with the standard extension to the product measure-space $\mathcal{F} \otimes \mathcal{F}$). It can be checked that $\gamma$ satisfies the conditions of the proposition. \hfill \Box
Proof of Prop. 8.6. First, we show that \( \mathcal{W}(\varepsilon) \leq \mathcal{W} \) for all \( \varepsilon > 0 \); For the rest of the proof, we will omit the argument \( \varepsilon \). For \( \delta > 0 \) arbitrarily small, we choose some \( \gamma \in \Gamma(\mu^\delta, \mu^\delta) \) within \( \delta \) of the infimum (8.1). Let \( f \) be given by (C.1). Then we have

\[
\mathcal{W} = \inf_{\gamma} \int \|p, p'\| \, d\gamma(p, p') \geq \int \|p, p'\| \, d\gamma(p, p') + \delta. \tag{C.2}
\]

Let us define the distance function

\[
\mathcal{D}(p, p') := \sum_{C, C' \in C} \min_{q \in C, q' \in C'} \mathcal{D}(q, q'). \tag{C.3}
\]

We observe \( \mathcal{D} \) is everywhere a lower bound for \( \mathcal{D} \); therefore,

\[
\int \|p, p'\| \, d\gamma(p, p') \geq \int \mathcal{D}(p, p') \, d\gamma(p, p'). \tag{C.4}
\]

Letting \( \mathbf{w}^{\text{LOWER}} := \{w^{\text{LOWER}}_a\}_{a \in A} \), note that

\[
\int \mathcal{D}(p, p') \, d\gamma(p, p') = \sum_{C, C' \in C} \min_{q \in C, q' \in C'} \|q, q'\| \int_{p \in C, p' \in C'} \, d\gamma(p, p') = \sum_{u \in V^+, u' \in V^+} w^{\text{LOWER}}(u, u') \tag{C.5}
\]

By definition, \( J(f; \mathbf{w}^{\text{LOWER}}) \) is no smaller than \( \mathcal{W} \). Combining these results we have that \( \mathcal{W} \geq \mathcal{W} + \delta \). The proof follows since the inequality holds for \( \delta \) arbitrarily small.

The proof that \( \mathcal{W} \geq \mathcal{W} \) is similar. Let \( f \) be the minimum-cost flow of \( \mathcal{N}^{\text{APPROX}} \) under edge weights \( \mathbf{w}^{\text{UPPER}} \); by definition, the cost of \( f \) is \( \mathcal{W} \). Recalling Remark C.2, let \( \gamma \) be any coupling of \( \mu^\delta \) and \( \mu^\delta \) which induces \( f \). Then

\[
\mathcal{W} = \inf_{\gamma'} \int \|p, p'\| \, d\gamma'(p, p') \leq \int \|p, p'\| \, d\gamma(p, p'). \tag{C.6}
\]

We define the distance function

\[
\mathcal{D}(p, p') := \sum_{C, C' \in C} \max_{q \in C, q' \in C'} \mathcal{D}(q, q'); \tag{C.7}
\]

\( \mathcal{D} \) is everywhere greater than \( \mathcal{D} \), so

\[
\int \|p, p'\| \, d\gamma(p, p') \leq \int \mathcal{D}(p, p') \, d\gamma(p, p'). \tag{C.8}
\]

By previous logic, it can be shown that

\[
\int \mathcal{D}(p, p') \, d\gamma(p, p') = J(f; \mathbf{w}^{\text{UPPER}}) = \mathcal{W}(\varepsilon). \tag{C.9}
\]
Combining these results proves the second part. \qed

**Proof of Prop. 8.7.** The result is simply a consequence of the fact (one can check) that for any $\epsilon > 0$, and $w^{\text{LOWER}}(\epsilon) = \{w_a^{\text{LOWER}}\}_{a \in A}$, $w^{\text{UPPER}}(\epsilon) = \{w_a^{\text{UPPER}}\}_{a \in A}$, we have $w_a^{\text{UPPER}} - w_a^{\text{LOWER}} \leq \epsilon$ for all $a \in A$. Let $f^*$ be the minimum-cost flow on $\mathcal{N}^{\text{APPROX}}$ with edge weights $w^{\text{LOWER}}$. Note that

$$
\mathcal{O}(\epsilon) = \min_{f \in \mathcal{N}^{\text{APPROX}}} J(\hat{f}; w^{\text{UPPER}}(\epsilon)) \\
\leq J(f^*; w^{\text{UPPER}}(\epsilon)) \\
= \sum_{(u, u') \in V} w_{(u, u')}^{\text{UPPER}} f^*(u, u') \\
\leq \sum_{(u, u') \in V} [w_{(u, u')}^{\text{LOWER}} + \epsilon] f^*(u, u') \\
= J(f^*; w^{\text{LOWER}}(\epsilon)) + \epsilon |\mu| = \mathcal{O}(\epsilon) + \epsilon |\mu|.
$$

(C.9)

### C.2 Reimann Approximation of Road Device Costs

**Proof of Lemma 8.14.** We give the proof only for $r \in S$; the proof for $r \in D$ is by identical logic. Since $g_r$ is parted [e.g., between $k_r - 1$ and $k_r$], we can restrict the ranges of the sums in (8.22) to obtain

$$
J_r(f^*; w^{\text{PATH}}) = \sum_{k=0}^{k_r-1} \epsilon_r f^*(u_r^{k+1}, u_r^k) + \sum_{k=k_r}^{N-1} \epsilon_r f^*(u_r^k, u_r^{k+1}).
$$

(C.10)

Combining the parted-ness of $g_r \in S$ with the flow conservation constraints (7.7), we obtain a recursive system

$$
f^*(u_r^k, u_r^{k-1}) = f^*(u_r^{k+1}, u_r^k) + b(u_r^k), \quad \text{for } k = 2, \ldots, k_r - 1, \text{ and } (C.11)
$$

$$
f^*(u_r^k, u_r^{k+1}) = f^*(u_r^{k-1}, u_r^k) + b(u_r^k), \quad \text{for } k = k_r + 1, N - 2. \quad (C.12)
$$

We can “unroll” each of the recursions (C.11) and (C.12) until we reach the part index $k_r$; since the supply $b(k_r)$ could be split between the backward and forward flows, at best we can write bounds

$$
\sum_{k' = k+1}^{k_r-1} b(u_r^{k'}) \leq f^*(u_r^{k+1}, u_r^k) \leq \sum_{k' = k+1}^{k_r} b(u_r^{k'}) \quad \text{for all } k < k_r, \quad (C.13)
$$

$$
\sum_{k' = k_r+1}^k b(u_r^{k'}) \leq f^*(u_r^k, u_r^{k+1}) \leq \sum_{k' = k_r+1}^k b(u_r^{k'}) \quad \text{for all } k \geq k_r. \quad (C.14)
$$

173
Substituting (C.13) and (C.14) in (C.10), and re-arranging the sums, we obtain bounds \( J_r \leq J_r \leq \bar{J}_r \), where

\[
J_r := \sum_{k'=1}^{k_r} b(u_r^{k'}) k' \epsilon_r + \sum_{k'=k_r+1}^{N-1} b(u_r^{k'}) (N - k') \epsilon_r, \\
\bar{J}_r := \sum_{k'=1}^{k_r} b(u_r^{k'}) k' \epsilon_r + \sum_{k'=k_r}^{N-1} b(u_r^{k'}) (N - k') \epsilon_r.
\]  

(C.15)

The two bounds have separation \( \bar{J}_r - J_r = b(u_r^{k_r}) [k_r + (N - k_r)] \epsilon_r = b(u_r^{k_r}) L_r \).

Since \( \varphi \) is Lipschitz by assumption, then

\[
b(u_r^{k}) = \epsilon_r \varphi_r(k \epsilon_r) + o(\epsilon_r) \quad \text{for} \quad k = 1, \ldots, N,
\]

(C.16)

and so \( J_r = \bar{J}_r + O(\epsilon) \). Substituting (C.16) into (C.15), as well as \( y(k) = k \epsilon_r \) and \( \Delta y = \epsilon_r \), we obtain

\[
J_r = O(\epsilon) + \sum_{k'=1}^{k_r} [\varphi_r(y(k')) \Delta y + o(\Delta y)] y(k') + \sum_{k'=k_r+1}^{N-1} [\varphi_r(y(k')) \Delta y + o(\Delta y)] (L_r - y(k')); \quad \text{(C.17)}
\]

(C.17) is a Reimann sum which can be written as (8.23).

(8.24) and (8.25) can be obtained in a similar fashion by substituting (C.16) into (C.13) and (C.14), respectively, for \( f(t_{\text{conn}}) \equiv f(u_r^0, u_r^0) \) and \( f(h_{\text{conn}}) \equiv f(u_r^N, u_r^{N+1}) \), then identifying the Reimann sums, and applying Definition 6.9. To obtain (8.25) also requires a change of variables \( y' = L_r - y \) and substitution by \( \chi_r \).
Bibliography


[7] RS Anderssen, RP Brent, DJ Daley, and PAP Moran. Concerning \( \int_0^1 \int_0^1 (x_1^2 + x_2)^2 \int_0^1 \int_1^1 \) and a taylor series method. SIAM Journal on Applied Mathematics, 30(1):22–30, 1976.


