To the Graduate Council:

I am submitting herewith a dissertation written by Alexander M. Melin. entitled "Bandwidth on Demand for Laser Communications Satellites." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Mechanical Engineering.

_____

Vijay Sekhar Chellaboina, Major Professor

We have read this dissertation
and recommend its acceptance:

Seddik Djouadi

_____

William Hamel

_____

Dongjun Lee

_____

Acceptance for the Council:

Carolyn R. Hodges

_____

Vice Provost and Dean of the
Graduate School

(Original signatures are on file with official student records.)

Bandwidth on Demand for Laser Communications Satellites

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Alexander M Melin
May 2008

Copyright © 2008 by Alexander M Melin.
All rights reserved.

Dedication

This dissertation is dedicated to my grandfather Bill Dieckman. Grandpa, I wish you were here to see this.

Acknowledgments

I would like to thank Dr. Vijay Chellaboina. By guidance and example, he has shown me how to truly think. I would like to thank Dr. Scott Erwin for placing me on this twisting path to my dissertation project. I would also like to thank Dr. Robby Robertson for convincing me to apply my ideas to laser communications. And not least, I would like to thank Dr. Seth Lacy for his guidance and support of this research summer after summer, and all of them for the wonderful opportunity to see the beautiful southwestern desert (and do a little work in the process). I want to thank Dr. Ajeet Kamath for being a great office mate and mentor.

I would also like to thank my family. I would like to thank Paul Blackford for having a robotic arm in his basement which inspired me to be an engineer before I was even old enough to know what an engineer is. I would also like to thank my grandparents for emparting to me the wisdom of there years. Finally, I would like my parents Mike and Sharon and my brother and sister Isaac and Lydia. Without their love and support I would never have made it this far.

Abstract

Narrow beamwidth satellites such as laser communication or imaging satellites present a unique problem in control and scheduling. The narrow beamwidth limits the sources of information available to the satellite. This is especially true for laser communication satellites who can only point the laser at one user at a time. For the most part, this restricts the use of laser communications satellites to fixed connections or dedicated temporary connections. However, some classes of users can benefit from the speed and security of the lasercom connection. For example, a Globalhawk unmanned aerial vehicle with time sensitive battlefield imaging would be in this class. For this class of users a lasercom bandwidth-on-demand system would allow for high speed, secure, timely data transfer capability. The consequence of this narrow beamwidth is that the switching time for the system can no longer be ignored. In this dissertation, we will formulate the bandwidth-on-demand lasercom problem mathematically and determine several objective functions for improving the performance and quality of service. Next, we will derive equations that describe the limiting behavior of this system from a queueing standpoint. Next, we will also analyze the performance of the proposed objective functions, and look at real time solution methods and their performance. Next, we present a new algorithm for solving the traveling salesman problem that is easily applied to the systems such a bandwidth-on-demand lasercom, whose behavior cost cannot be described by a cost matrix that characterizes a standard traveling salesman problem. Finally, we present a system for emulating global coverage with a constellation of narrow beamwidth satellites such as imaging or laser communications satellite.

Contents

1 Introduction and Background **1**
- 1.1 Introduction . . . 1
- 1.2 Background . . . 1
  - 1.2.1 Free Space Laser Communications . . . 1
  - 1.2.2 Dynamic Traveling Salesman Problem . . . 3
  - 1.2.3 Dynamic Traveling Repairman Problem . . . 5
- 1.3 Contributions . . . 7

2 Lasercom Queueing System Model Development **8**
- 2.1 Introduction . . . 8
- 2.2 Notation and Standard Definitions . . . 8
- 2.3 Problem Definition . . . 11
- 2.4 Performance Measures . . . 13
  - 2.4.1 Variation of Wait Time (VW) Service Rule . . . 16
  - 2.4.2 Nearest Insertion Service Rule . . . 17

3 Analysis of Queue Behavior **19**
- 3.1 Introduction . . . 19
- 3.2 Mathematical Preliminaries . . . 19
- 3.3 Maximum Connection Time . . . 20
- 3.4 Birth-Death Queueing System in Equilibrium . . . 22
  - 3.4.1 General Solution . . . 22
  - 3.4.2 M/M/1 Queueing System . . . 23
- 3.5 Average Bandwidth . . . 24
- 3.6 Stability Analysis . . . 24
- 3.7 Wait Time . . . 26
- 3.8 Wait Time Variance . . . 29

4 Performance Analysis **31**
- 4.1 Convergence of Solutions . . . 31
- 4.2 Stationary Target Service Rules . . . 35
  - 4.2.1 Zero Switching Time . . . 35
  - 4.2.2 Medium Switching Time . . . 35
  - 4.2.3 Large Switching Time . . . 42
  - 4.2.4 Very Large Switching Time . . . 42
  - 4.2.5 General Comments . . . 42
- 4.3 Extended Lin-Kernighan . . . 45

4.3.1 TSP Heuristic Basics . . . 45
4.3.2 Tour Optimality . . . 47
4.3.3 Lin-Kernigan . . . 47
4.3.4 Extended Lin-Kernighan . . . 47
4.3.5 Relationship to Lin-Kernighan . . . 48
4.3.6 Reducing Running Time . . . 48

5 Low-Earth Orbit Satellite Constellations **51**
5.0.7 Low-Earth Orbit Satellite Constellations . . . 51

6 Conclusion **59**
6.1 Conclusion . . . 59
6.2 Contributions . . . 59
6.3 Future Work . . . 60

Bibliography **61**

Appendix **67**

Vita **83**

List of Tables

6.1 Solutions of TSP instances with Concorde. 78

List of Figures

2.1 System time breakdown for first come first served service rule 14

3.1 Average Bandwidth 25
3.2 Stability region for varying switching time. 25
3.3 Comparison of the calculated and simulated average queue size for various service disciplines. 27
3.4 Comparison of the calculated and simulated average wait time for various service disciplines. 27
3.5 Comparison of the new expected value for the average wait time and the Bertsimas and Van Ryzin bounds. 28
3.6 Comparison of the calculated and simulated wait time variance for various service disciplines. 30

4.1 Low Traffic Average Queue. 32
4.2 Low Traffic Average Queue. 32
4.3 Medium Traffic Average Queue. 33
4.4 Medium Traffic Average Queue. 33
4.5 Heavy Traffic Average Queue. 34
4.6 Heavy Traffic Average Queue. 34
4.7 Average wait time for three service disciplines with zero switching time. 36
4.8 The percent decrease in the wait time from the first come first served service rule. . 36
4.9 Variance of the wait time with zero switching time. 37
4.10 The percent decrease in wait time variation from the first come first served service rule. 37
4.11 Average wait time for three service disciplines with medium switching time. 38
4.12 The percent decrease in the wait time from the first come first served service rule. . 38
4.13 Variance of the wait time with medium switching time. 39
4.14 The percent decrease in wait time variation from the first come first served service rule. 39
4.15 Average wait time for three service disciplines with large switching time. 40
4.16 The percent decrease in the wait time from the first come first served service rule. . 40
4.17 Variance of the wait time with large switching time. 41
4.18 The percent decrease in wait time variation from the first come first served service rule. 41
4.19 Average wait time for three service disciplines with large switching time. 43
4.20 The percent decrease in the wait time from the first come first served service rule. . 43
4.21 Variance of the wait time with large switching time. 44

4.22 The percent decrease in wait time variation from the first come first served service rule. 44
4.23 Upper bound on the initial nearest neighbor tour quality. 46

5.1 A complete bipartite graph, a matching on a bipartite graph and the equivalent edge matrix 52
5.2 A minimum line covering on an edge matrix 53
5.3 Augmented edge matrix with the new minimum line cover 54
5.4 GlobalStar Satellite Constellation 57

6.1 Knight's tour discovered by Euler. 73
6.2 Upper bound on the initial nearest neighbor tour quality. 79
6.3 A 2-opt move 82

Chapter 1

Introduction and Background

1.1 Introduction

In this chapter we will discuss previous work related to the control and routing of free space laser communications (lasercom). First we will discuss the history of laser communications including device development and hardware testing. We will also discuss the physical issues that make the system unique from a network standpoint. Next, we will discuss the traveling salesman problem, the dynamic traveling salesman problem and their relationship to optimizing the lasercom system. Then, we will look at the problem formulation of the dynamic traveling salesman problem and work done on the solution of that problem, along with important differences between the traveling salesman problem and the dynamic traveling salesman problem. Finally, we will look at the application of the dynamic traveling salesman problem to lasercom and other related satellite systems and the implications that this has on networking theory.

1.2 Background

We will begin motivating the problem by a short background on the developments in free-space laser communications to date.

1.2.1 Free Space Laser Communications

Free space laser communications is being investigated by numerous corporations and government agencies. This is due to the high data transmission rates, low power requirements for long range communications, and security due to the difficulty of intercepting the signal. This is a relatively new field of study however, because of the large technical challenges to overcome. NASA began conducting system studies and technology development in 1989 and finally conducted a successful test of a ground to space free space laser system in 1992 during the Galileo Optical Experiment (GOPEX). This experiment did not establish a data link, but showed the feasibility of a spacecraft detecting a beam from earth [1]. The first dual link experiment was accomplished in 1995 during NASA's Ground/Orbiter Laser Demonstration (GOLD) [2]. In late 2001, the European Space Agency (ESA) performed the first satellite to satellite link with their Semiconductor-laser Intersatellite Link Experiment (SILEX) [3]. Then, in 2005 the Japanese OICETS satellite became the second link the optical data relay by connecting their satellite to the SILEX satellite.

The U.S. military is in the process of augmenting its RF communications system with a combination of lasercom and extra high frequency (EHF) communications. This program is known as the

Transformational Satellite Communications System (TSAT). It is being developed by the Air Force Research Lab (AFRL) in conjunction with Boeing and various other contractors. In 2006 Boeing successfully demonstrated a satellite to satellite link with TSAT in a ground based simulation of the space environment [4]. This system is scheduled to begin launching in 2011 [5]. Currently TSAT will be used as an information backbone that can be reconfigured depending on the geographic locations of the terminals. It will be used in two ways, first, to connect geographically separate user networks as a virtual private network, second to connect a mobile ad-hoc network that has separated into two or more networks that can no longer physically connected [6].

The problem proposed in this paper is an extension of the TSAT system capability. In addition to providing a reconfigurable network backbone, we propose to use the system as a bandwidth on demand capability for classes of mobile users that need high priority access to the network is needed. This class would include users such as the GlobalHawk unmanned aerial vehicle surveillance platform.

We will restrict ourselves to the terminal to satellite connection. We will not include the satellite to satellite backbone, or the architecture to transfer the data from the satellite to the intended recipient. These network architecture problems are similar to standard internet architectures on the ground and in RF satellite communications and have been studied thoroughly in the past [7–20]. Instead, the system will be analyzed from a global perspective for behavior and performance.

This problem is unique in the terminal to satellite connection due to the physical properties of the communications link. These properties cause the switching dynamics of the system to be different that a normal network. For the satellite to track and communicate with an individual terminal, it must be aware that a terminal needs a link established, i.e. see the terminal's laser on its wide field sensor. The satellite must then search the approximate area where the user is located with it's own laser until the terminal sends back a signal that a connection has been established. Then, with both lasers pointed at the other, the communications link can be established. This causes the switching time for this network to be much higher than other systems.

Since the lasercom satellites will be stationed in geosynchronous orbit at an altitude of 35,786 km above the mean ocean level, the time for a signal to travel from the satellite to the earth is approximately 0.1194 seconds, with a round trip of 0.2388 seconds. This does not include the time the satellite takes to locate the terminal so this places a lower bound on the time to switch between users at 0.2388 seconds. This is significantly higher than the switching time for any other communications network, radio frequency (RF) satellite, or ground based. As most current network routing protocols depend on very small packet sizes with a large amount of switching between terminals, the current switching protocols would cause large inefficiencies in the system performance. The analysis of these systems ignores the switching time. The switching time for a lasercom system dominates the time to send a typical length IP packet. For example, an IP packet has 160 bits, while a lasercom transmitter developed by Ball Aerospace has a transmission rate of 40 Gbps. With transmission time for one IP packet of 4e-9 seconds, and a minimum switching time of 0.2388 seconds, the switching time is greater than the transmission time by a factor of *60 million*. This is the equivalent bandwidth of 670 bits/second, a prohibitive loss in bandwidth. In the time that it takes to switch from one terminal to another 9.552 Gigabits can be sent which is approximately 10.3 hours of AVI video. This forces a new thinking about network routing protocols.

Another difference between a bandwidth on demand lasercom system and a conventional system is the high degree of uncertainty about the terminal locations when the terminals are moving. This uncertainty causes an additional increase in the switching time. This arises because of the narrow field of view of the laser. When there is no connection between a terminal and the satellite, the satellite only has a rough location for the terminal from the wide field of view sensor. If the satellite

has been connected to the terminal recently, the search area can be reduce by saving the terminal's previous location, heading, and speed at the end of the last connection.

This uncertainty in location leads to an increase in the switching time. For a bandwidth on demand system to be practical, some method must be determined to reduce the impact of the switching delay.

To generalize, the decision that the satellite makes is the length of time to hold specific connection and what order to serve the terminals. As we will show in Chapter 3, the most efficient scenario is a burst transmission where the terminal sends all data before the satellite switches to the next terminal. The other aspect of the switching problem is the order of service.

Stated simply, this problem is:

- Choose the order of service for the terminals, so that some metric is minimized or maximized

These metrics will be discussed in detail in Chapter 2. This problem can be thought of as a variation of the traveling salesman problem (TSP) if the terminals are static. When the terminal in motion which causes uncertainty to enter into the system, it can be described by the dynamic traveling repairman problem.

1.2.2 Dynamic Traveling Salesman Problem

Next, we will next look at the dynamic traveling salesman problem (DTSP) and the dynamic traveling repairman problem (DTRP) which are sub-problems of the dynamic vehicle routing problem (DVRP). The difference between the two formulations is the number of vehicles allowed and the time each vehicle stops at each vertex. The dynamic versions of the TSP capture the uncertainty and dynamic nature of the lasercom system while allowing for a degree of flexibility in design in the form of performance metrics that cover a wide range of potential uses. This requires a generalized knowledge of the behavior of the DTSP under differing conditions. Unfortunately, despite the fact that many practical systems are dynamic in nature and are best described by the DTSP, there has been a surprisingly small amount of work on the dynamic traveling salesman problem due to the difficulty in analyzing solving and analyzing the problem.

Psaraftis, [21] and [22], was one of the first to study dynamic versions of the traveling salesman problem and the vehicle routing problem. The following definition of the dynamic traveling salesman problem is from his 1988 paper, and is the first mathematical definition of the problem.

Problem 1.2.1 (Dynamic Traveling Salesman Problem [21]) *Let G be a complete graph* of m nodes. Assume that demands for service at each node is a Poisson process with parameter λ and are independent of one another, whose probability distribution function is given by:*

$$P[(N(t+\tau) - N(t)) = k] = \frac{e^{-\lambda\tau}(\lambda\tau)^k}{k!} \tag{1.1}$$

where $(N(t+\tau) - N(t))$ describes the number of events in the time interval $(t, t+\tau]$. Furthermore, these demands are to be serviced by a salesman who takes a (known) time of t_{ij} to travel from node i to node j of G, and spends a (known) time of t_0 servicing each demand (on location). Assume that at time zero the salesman is at node 1. Determine the "optimal" routing policy in the sense that,

(a) maximize the average demands serviced per unit time

(b) minimize the average, over all demands, the expected time from the appearance of a demand until its service is completed.

*See Section 2.2 for a definition.

In [21], Psaraftis also lists twelve differences between the dynamic and static versions of the problem. These are important in understanding what techniques can be transferred from the static problem to the dynamic traveling salesman problem in addition to practical consideration on the implementation of real-time routing policies.

Time dimension is essential. In a static routing problem, the time dimension may or may not be important. In the dynamic counterpart, time is always essential. The dispatcher must as a minimum know the position of all vehicles at any given point in time and particularly when the request for service or other information is received by the dispatcher.

The problem may be open-ended. The process is often temporally bounded in a static problem. The routes start and end at the depot. In a dynamic setting the process may very well be unbounded. Instead of routes one considers paths for the vehicles to follow.

Future information may be imprecise or unknown. In a static problem all information is assumed to be known and of the same quality. In a real-life dynamic routing problem, the future is almost never known with certainty. At best probabilistic information about the future may be known.

Near-term events are more important. Due to the uniformity of the information quality and lack of input updates all events carry the same weight in a static routing problem. Whereas in a dynamic setting it would be unwise to immediately commit vehicle resources to long-term requirements. The focus of the dispatcher should therefore be on near-term events when dealing with a dynamic routing problem.

Information update mechanisms are essential. Almost all inputs to a dynamic routing problem are subject to changes during the course of operation. It is therefore essential that information update mechanisms are integrated into the solution method. Naturally, information update mechanisms are not relevant within a static context.

Re-sequencing and reassigning decisions may be warranted. In dynamic routing, new inputs may imply that previous decisions made by the dispatcher become suboptimal. This forces the dispatcher to re-route or reassign vehicles in order to respond to the new situation.

Time constraints restrict solution methods. In static settings the dispatcher may afford the luxury of waiting for a few hours in order to get a high quality solution, in some cases even an optimal one. In dynamic settings this is not possible, because the dispatcher wishes to know the solution to the current problem as soon as possible (preferably within minutes or seconds). The *running-time* constraint implies that re-routing and reassignments are often done by using local improvement heuristics like insertion and k-interchange.

Indefinite deferment mechanisms are essential. Indefinite deferment means the eventuality that the service of a particular demand will be postponed indefinitely because of that demand's unfavorable geographical characteristics relative to the other demands. This problem could, for instance, be alleviated by using time window constraints or by using a nonlinear objective function penalizing excessive wait.

Objective function may be different. Traditional static objectives such as minimization of the total distance traveled or the overall duration of the schedule might be meaningless in a dynamic setting because the process may be open-ended. If no information about the future inputs is available, it might be reasonable to optimize only over known inputs. Some systems

could also use nonlinear objective functions in order to avoid undesirable phenomena such as the above mentioned deferment.

Time constraints may be different. Time constraints such as latest pickup times tend to be softer in a dynamic routing problem than in a static one. This is due to the fact that denying service to an immediate demand, if the time constraint is not met, is usually less attractive than violating the time constraint.

Flexibility to vary vehicle fleet size is lower. In static settings the time gap between the execution of the algorithm and the execution of the routes usually allows adjustments of the vehicle fleet. However, within a dynamic setting the dispatcher may not have instant access to backup vehicles. Implications of this may mean that some customers receive lower quality of service.

Queueing considerations may become important. If the rate of customer demand exceeds a certain threshold, the system will become congested and the algorithms are bound to produce lower quality results. Although vehicle routing and queueing theory are well-studied, the effort to combine these theories has been scant.

As we will see later, queueing consideration are indeed important in this problem, especially when applied to communications networks.

1.2.3 Dynamic Traveling Repairman Problem

The dynamic traveling repairman problem (DTRP) is the extension of the traveling salesman problem that most closely describes the laser communications system under consideration. The dynamic traveling repairman problem was introduced by Bertsimas and Van Ryzin in [23] and was initially motivated by Psaraftis's [21] work on the DTSP. Bertsimas and Van Ryzin's work is by far the most extensive and mathematically concise work on dynamic vehicle routing problems.

Bertsimas and Van Ryzin define the DTRP as follows:

Problem 1.2.2 *Dynamic Traveling Repairman Problem*

- *A repairman (or a vehicle/server) travels at unit velocity in a bounded convex region $\mathcal{A}$ of area A.*
- *All demands are dynamic and arrive in time according to a Poisson process with the intensity parameter λ. The locations of the demands are independently and uniformly distributed in $\mathcal{A}$.*
- *Each demand requires an independently and identically distributed amount of on-site service time with mean duration $\overline{s}$ and second moment $\overline{s}^2$. The fraction of time that the server spends on on-site servicing the demands is denoted ρ. For stable[†] systems $\rho = \lambda\overline{s}$.*
- *The system time, T_i, of demand i is defined as the elapsed time between the arrival of demand i and the time the server completes the service of the demand. The steady-state system time denoted by T is defined by $T = \lim_{i=\infty} E[T_i]$.*
- *The waiting time, W_i, of demand i is defined as the time elapsed from the arrival of the demand until the service starts. Thus, $T_i = W_i + s_i$. The steady-state waiting time, W, is defined as $W = T - \overline{s}$.*

[†]See Section 2.2 for a definition

- *The objective is to design a routing policy that minimizes the system time T.*

While this problem resembles traditional queueing theory, it does not apply in this instance due to the fact that the system time includes the travel times which are not independent variables. In [23], the problem is divided into two cases, light traffic ($\lambda \to 0$) and heavy traffic ($\lambda \to 1$), and lower bounds for different routing policies are compared.

When the system has light traffic a lower bound on the system time was determined to be

$$T^* \geq \frac{E[||X - x^*||]}{1-\rho} + \frac{\lambda \bar{s}^2}{2(1-\rho)} + \bar{s}. \tag{1.2}$$

If $\mathcal{A}$ is square, $E[||X - x^*||] = 0.383\sqrt{A}$. The bound grows with larger values of ρ.

When the system has heavy traffic, there exists a constant $\gamma = \frac{2}{3}\sqrt{2}\pi \approx 0.266$ such that:

$$T^* \geq \gamma^2 \frac{\lambda A}{(1-\rho)^2} - \frac{1-2\rho}{2\lambda}. \tag{1.3}$$

This bound grows with a factor of $(1-\rho)^{-2}$. In traditional queueing systems, the system time grows with a factor of $(1-\rho)^{-1}$. This difference stems from the geometry of the system. In [23], the authors conjecture that γ is not tight.

The authors also analyze different routing policies as described below.

First Come First Served (FCFS). The demands are served in the order in which they are received by the dispatcher.

Stochastic Queue Median (FCFS-SQM). The FCFS-SQM policy is a modification of the FCFS policy. According to the FCFS-SQM policy the server travels directly from the median of the service region to the location of the demand. After the service has been completed, the server returns to the median and waits for the next demand.

Nearest Neighbor (NN). After completing service at one location the server travels to the nearest neighboring demand.

Traveling Salesman Problem (TSP). The demands are batched into sets of size n. Each time a new set of demands has been collected, a TSP is solved. The demands are served according to the optimal TSP tour. If more than one set exists at a time, the sets are served in an FCFS manner.

Space Filling Curve (SFC). The demands are served as they are encountered during repeated clockwise sweeps of a circle $\mathcal{C}$ that covers the service region.

Partitioning Policy (PART). The service region $\mathcal{A}$ is partitioned into m^2 smaller subregions in which the demands are served using an FCFS policy. When no more demands are left in a subregion, the server travels to the next subregion.

The FCFS-SQM policy can be shown to give asymptotically optimal performance in light traffic. The simulations show the SFC and NN policies are efficient in both light and heavy traffic. Furthermore, both of these policies are non-parametric so they are more stable when there are large variations in λ.

In [24], Bertsimas and Van Ryzin generalize the findings of their first paper. They provide bounds and propose a number of new policies for both the un-capacitated m-server case and for

the case with m servers each with a capacity of q. In this paper each vehicle travels at a constant velocity v.

For light traffic, the lower bound on the system time is given by:

$$T^* \geq \frac{1}{v} E[\min_{x_0 \in \mathcal{D}^*} ||X - x_0||] + \overline{s}. \tag{1.4}$$

This is the expected travel time plus the on-site service time.

For heavy traffic, the lower bound becomes:

$$T^* \geq \gamma^2 \frac{\lambda A}{m^2 v^2 (1-\rho)^2} - \frac{\overline{s}(1-2\rho)}{2\rho}. \tag{1.5}$$

The SQM policy for the un-capacitated case can be extended to m servers by locating them at m median locations in the region $\mathcal{A}$. New demands are assigned to the nearest median location and its corresponding server. It is shown that this policy is optimal for light traffic. For heavy traffic, the authors also propose two policies. The first is the Random Assignment policy (RAμ), in which the Poisson process is divided into m Poisson subprocesses. Each server is assigned to a subprocess and demands are served using a heavy-traffic policy μ. The RAμ policy has a constant factor guarantee, but the system time increases by m. The other heavy traffic policy proposed is the G/G/m version of the TSP. In this policy, the demands are collected into sets of size n. An optimal TSP is found each time a new set is collected. The optimal tours are then served according to a FCFS scheme. It is shown that for a modified version of this policy, the system time can be made independent of m.

For simplicity, we will not discuss the capacitated vehicle case.

In their papers, Bertsimas and Van Ryzin apply the routing policies to minimize the overall distance that the repairman travels with the cost function given by the travel distance. We will approach the problem by using the system time, and wait time as functions of the route as our cost functions. This is especially applicable to the laser communications problem with stationary terminals, as the cost to switch between terminals (analogous to the distance) is a constant.

1.3 Contributions

The main contributions of this dissertation are a mathematical definition of the bandwidth-on-demand laser communications problem, both from an system perspective and a solution perspective. Additionally, we have defined three new cost functions that show marked improvement in reducing the system time and wait time over the traditional service disciplines such as first-come first-served. Next, we have analyzed the system from a queueing theory perspective and derived equations that describe the average behavior of the system wait time and system time along with the variance of those parameters. We also show that these equations more accurately describe the average system behavior than the best previously known bounds derived by Bertsimas and Van Ryzin. Next, we show that the cost functions presented in this dissertation provide better performance than the standard service disciplines. Next, we present a new algorithm for solving the traveling salesman problem that is easily applied to the systems such a bandwidth-on-demand lasercom, whose behavior cost cannot be described by a cost matrix that characterizes a standard traveling salesman problem. Finally, we present a system for emulating global coverage with a constellation of narrow beamwidth satellites.

Chapter 2

Lasercom Queueing System Model Development

2.1 Introduction

A satellite laser communications system is very similar to a land based communications network. The main differences arise from the physical distance between the satellite and the terminals and the narrow beamwidth of a lasercom satellite. A traditional radio-frequency (RF) satellite communication system such as MILSATCOM must deal with the transmission delay arising from the distance between links. However, RF satellites can send and receive transmissions from multiple users simultaneously due to the wide broadcast and reception cone of these satellites. Laser communications systems, on the other hand, must physically move the laser to the receiver on the target to establish a connection. This causes large switching delays, especially with customers that are not stationary.

Other than the switching delay, this system is nothing more that a traditional queueing system, however. For simple queueing systems there are many well known analytical results that can be applied to the laser communications system, however, the switching delay, especially for moving targets complicates the analysis of this system. For simplicity we will only consider stationary targets, but we will briefly discuss issues involved with moving targets. In Section 2.2 some present some basic notation and standard definitions. In Section 2.3 we will develop various formulations of the problem. Section 2.4 will present various objective functions that will be used in Chapters 3 and 4.

2.2 Notation and Standard Definitions

Let $\mathbb{R}$ (resp., $\mathbb{Z}$) denote the set of real numbers (resp., positive) integers, let $\mathbb{R}^{m\times n}$ (resp., $\mathbb{Z}^{m\times n}$) denote the set of real numbers (resp., positive) matrices, and let $\mathbb{N}$ (resp., $\mathbb{N}^+$) denote the set of natural numbers (resp., positive natural numbers). Let $G = (V, E)$ denote a graph where V is the set of vertices and E is the set of edges. Furthermore E can be represented by the connectivity (or adjacency) matrix given by

$$E_{ij} = \begin{cases} 1, & \text{if } (i,j) \text{ is an edge,} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

Let a *complete graph* be a graph in which all vertices are connected to all other vertices. Let $G = (V, E, W)$ be a weighted graph with $W \in \mathbb{R}^{n\times n}$ denoting the weight matrix associated with

each edge. A *circuit* of a graphs is a series of vertices and edges in which the first and last vertex are coincident. A *Hamiltonian circuit* also known as a *tour* is a circuit that passes through each vertex only once.

The laser communications system, like any communications system, is at heart governed by statistical processes. People don't use the telephone on a predefined schedule, they use it randomly, at the times that they need. The two variables in a communications system that are modeled by random variables are the arrival of the requests for service, and the duration of the service. The majority of communications systems use a Poisson process to describe the arrival rate and service times. This has been shown to be a very accurate long term description of arrival and departure of customers in most communication systems.

The Poisson distribution is defined as:

Definition 2.2.1 (Poisson Distribution) *[25] Given a rate parameter λ and a time interval $(t, t+\tau]$, the **Poisson distribution** is given by*

$$P[(N(\tau+t)-N(\tau))=k]=\frac{e^{-\lambda t}(\lambda t)^k}{k!}, \quad k=0,1,\ldots,\infty, \tag{2.2}$$

where $N(\tau+t)-N(\tau)$ describes the number of events in the time interval $(t, t+\tau]$.

The notation $P_k(t)$ will be used as a shorthand description of the Poisson probability distribution function* given by 2.2. The parameter λ is the average arrival rate of customers into the system. Intuitively, the average number of arrivals in an interval t is given by λt. More precisely, defining K to be the number of arrivals in an interval of length t and requiring that K be nonnegative, the expected value of K is given by:

$$\begin{aligned} E[K] &= \sum_{k=0}^{\infty} kP_k(t) \\ &= e^{-\lambda t}\sum_{k=0}^{\infty} k\frac{(\lambda t)^k}{k!} \\ &= e^{-\lambda t}\sum_{k=1}^{\infty} \frac{(\lambda t)^k}{(k-1)!} \\ &= e^{-\lambda t}\lambda t\sum_{k=0}^{\infty} \frac{(\lambda t)^k}{k!} \\ &= \lambda t. \end{aligned} \tag{2.3}$$

Since λ is the rate of arrival, the expected number of arrivals in a time interval of length t is simply λt. Additionally, the variance of the Poisson distribution is given by $\sigma_P^2 = \lambda t$. Next we will define the Poisson process which describes the arrival times due to a Poisson distribution.

Definition 2.2.2 (Poisson Process) *[25] A **Poisson process** is a process satisfying the following properties*

i) The number $N(t+\tau)-N(t)$ of points in an interval $(t, t+\tau]$ of length τ is a Poisson random variable with parameter $\lambda\tau$ that follows a Poisson distribution.

*See Appendix I for basic statistical notation and definitions.

ii) If the intervals $(t_1, t_1 + \tau_1]$ and $(t_2, t_2 + \tau_2]$ are non-overlapping, then the random variables $N(t_1 + \tau_1) - N(t_1)$ and $N(t_2 + \tau_2) - N(t_2)$ are independent

While the Poisson process determines the number of arrivals in a time interval, the exponential distribution describes the inter-arrival time of customers into the system. The *exponential distribution* is given by:

$$P[X \leq x] = 1 - e^{-\lambda x}, \tag{2.4}$$

with a probability density function given by:

$$f_X(x) = \begin{cases} \lambda e^{-\lambda x}, & 0 \leq x < \infty, \\ 0, & x < 0. \end{cases} \tag{2.5}$$

This distribution is related to the Poisson distribution in the following manner. If the inter-arrival times are given by

$$t_n = \tau_n - \tau_{n-1} = -\frac{1}{\lambda} \ln(y), \tag{2.6}$$

where y is a uniformly distributed random variable, then t_n is exponentially distributed and the arrival times τ_n follow a Poisson distribution[†]. The expected value for an exponential distribution is given by $E[x] = \frac{1}{\lambda}$ and the variance is given by $\sigma_E^2 = \frac{1}{\lambda^2}$.

A queue in a scientific sense, is a system which has an influx of users that require a certain amount of the system's resources before they leave.

Definition 2.2.3 (Queue) *[26] For every $n \in \mathbb{N}$, let t_n and S_n denote positive real-valued random variables with $t_{n+1} > t_n$ for all $n \in \mathbb{N}$. The sequence $\mathcal{T} = \{\tau_n : n \in \mathbb{N}_0\}$ is called **arrival point process** and $\mathcal{S} = \{s_n : n \in \mathbb{N}\}$ is the sequence of **service times.** Further choose a number k of servers and the **system capacity** C, with $k, C \in \mathbb{N} \cup \{\infty\}$. Let B be the **service discipline** or rule that determines the order that the members of the queue are served such as first come first served. Then the 5-tuple $(\mathcal{T}, \mathcal{S}, k, C, B)$ is called a **queue** or **queueing system**.*

In literature on queueing theory, a shorthand notation for a queue has been adopted. In general, the system capacity C and the service discipline B are ignored, and the arrival point process, service times, and number of servers are listed as $\mathcal{T}/\mathcal{S}/k$. The three types of arrival point processes and service times are Markov (M), deterministic (D), and general (G). The laser communications system with Poisson arrival rates, and exponential service times and one laser terminal would be denoted as an M/M/1 queue. This is the most common queue studied. No analytical solution is known for the most general queue G/G/n.

The most important quantity in a queueing system is the utilization factor. This parameter governs the behavior of the queue. It is also known as the traffic intensity parameter and from this one quantity many quantities such as the average number of users in the queue can be defined.

Definition 2.2.4 (Utilization Factor) *The **utilization factor** of the system ρ is defined by:*

$$\rho \triangleq \frac{\lambda}{\mu}. \tag{2.7}$$

We also want to determine is a queue is stable or unstable. In literature on queueing theory, the use of the word stable is not mathematically correct from a controls system standpoint, stability in this case refers to the boundedness of the queue. However, we will retain the common nomenclature from queueing theory for continuity.

[†]For a proof of this, see Appendix I.

Definition 2.2.5 (Stable Queue) *A queue is* ***stable*** *iff*

$$E[N] < \infty. \tag{2.8}$$

It is obvious from the definition of the utilization factor that if $\rho > 1$, then the arrival rate of customers into the queue is greater than the rate at which the customers can be serviced. This causes an ever increasing number of people in the queue. We will show in Chapter 3 that the inclusion of switching time into the queueing system will alter the definition of the utilization factor and consequently the stable operating range of a queueing system with switching time.

2.3 Problem Definition

We will look at three definitions of the laser communications system. First, we will define the system as a queue. This definition will be used to analyze the general behavior over the long term. Next, we will define the system as a variation of the linear programming problem. This definition is useful for short term analysis and the application of various service protocols to the real time control of the system. Finally, we will defined the system as an optimization on a graph. This definition, while not being as useful for analysis as the others, provides a link between different areas of mathematics that might lead to the application of tools from one area being applied to the other. But first we will describe the assumptions and notation used to define the system.

The laser communications system behavior if governed by four primary variables, the inter-arrival times of customers, the time it takes to serve each customer, the time to switch from one customer to another, and the number of servers. For this system we will assume that there is only one server.

Definition 2.3.1 (Satellite Terminal) *Let S be a satellite laser communications terminal located in geosynchronous orbit. This terminal will be considered a single server with a constant rate of data transfer r. The time to switch from one terminal is assumed constant and denoted by κ_s. We assume that the satellite's connection to the rest of the network has sufficient capacity to handle all data transactions.*

Definition 2.3.2 (Earth Terminals) *Let $\mathcal{C}$ be a set of stationary laser communications terminals located on earth. Each terminal will also be referred to as a customer. C_n will denote the n^{th} terminal to request service.*

τ_n will denote the arrival time for C_n with an inter-arrival time of

$$t_n = \tau_n - \tau_{n-1}. \tag{2.9}$$

Each terminal has an unknown location in a closed convex region $\mathcal{A}$. The amount of service time each terminal requires is denoted by s_n.

Let the waiting time for each terminal C_n be denoted by w_n and the system time for each terminal by T_n where

$$T_n \triangleq w_n + s_n. \tag{2.10}$$

The waiting time is divided into the the waiting time due to the switching time κ_s and the waiting time due to the service times of prior demands w^s where

$$w_n = \kappa^s + w_n^s, \tag{2.11}$$

and

$$T_n = w_n + s_n = \kappa^s + w_n^s + s_n. \tag{2.12}$$

Figure 2.1 shows the arrival, wait time due to the service of prior demands, switching time, and service time for a standard first come first served service rule.

Given the definitions of the earth terminals and the laser communications satellite, we can define the problem as a queue.

Problem 2.3.1 *Consider a set of terminals on the earth $\mathcal{C}$ given by Definition 2.3.2. We will assume that the arrival times of requests for service, denoted by $\mathcal{T} = \{\tau_n : n \in \mathbb{N}\}$, are given by a Poisson process with rate parameter λ. Additionally, we will assume that the service times for the terminals, denoted by $\mathcal{S} = \{s_n : n \in \mathbb{N}\}$, will be exponentially distributed with parameter μ. Consider a single server given by Definition 2.3.1 with a capacity or rate of data transfer given by r. Determine the order of service $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ that minimizes a performance metric $P = f(\mathcal{T}, \mathcal{S}, \lambda, \mu, r, \kappa^s)$.*

The order of service will be determined by the service discipline used. Because of the statistical nature of this problem formulation, it is useful for finding the expected values of the quantities and performance metrics. In this case, various service disciplines will be chosen and the long term performance of the system will be found through simulation. In some special cases, because of the simplicity of the service discipline, analytic results can be found for the system performance.

While this model can be used to describe the short term behavior of the queue, it can only do so for the simplest systems. For a stable queueing system, at any given time there are $N(t)$ customers in the queue. To describe the service discipline used to decide the order of service for these customers, we will turn to a variation of the linear programming model.

Problem 2.3.2 (Linear Programming) *Consider a fully connected graph $\mathcal{G}(\mathcal{U}, \mathcal{E})$ which represents all the users requesting connections.*

- *Let x_{ij} $(i \neq j)$ be a binary variable where $x_{ij} = 1$ only if the arc between users i and j are in the route.*
- *Let $\mathcal{X} \triangleq \{X \in \mathbb{R}^{n \times n} : x_{ij} = \{0, 1\}, i \neq j\}$ be the set of all possible routes.*
- *Define the set of earth terminals as $\mathcal{C}$, and the lasercom satellite as S*
- *Define $\mathcal{V} \triangleq \{v_i \in \mathcal{C} : i \in \mathbb{Z}\} \subset \mathcal{X}$ be the users in the laser's route.*
- *Let $\mathcal{K} = \{\kappa_{ij}\}$ where $\mathcal{K} \in \mathbb{R}^{n \times n}$ are the times associated with switching from the i^{th} user to the j^{th} user at time t.*
- *Let $W = \{w_{ij}(t, x_{ij}, \kappa_{ij}^s)\}$ where $W \in \mathbb{R}^{n \times n}$ is a nonnegative matrix denoting the cost associated with switching from the i^{th} user to the j^{th} user.*
- *Let τ_i, the arrival time of the i^{th} customer, be generated according to a Poisson process with parameter λ.*

This leads to the problem formulation given by,

$$\begin{aligned}
\textit{minimize} \quad & \textstyle\sum_{i \neq j} && w_{ij}(t, x_{ij}, \kappa_{ij}^s) x_{ij} && (2.13)\\
\textit{s. t.} \quad & \textstyle\sum_{j=1}^{n} && x_{ij} = 1 \quad \forall i \in \{1, \ldots, n\} && (2.14)\\
& \textstyle\sum_{i=1}^{n} && x_{ij} = 1 \quad \forall j \in \{1, \ldots, n\} && (2.15)\\
& \textstyle\sum_{i,j \in K} && x_{ij} \leq |K| - 1 \quad (K \subset \mathcal{V} \backslash v_1, |K| \geq 2). && (2.16)
\end{aligned}$$

This model is only valid if there exists a well defined cost matrix W. While this can be used for optimizing certain parameters in the lasercom system, it does not hold when trying to minimize the wait time or wait time variance. These quantities not only rely on each user's position in the tour, but also on the permutations of the users that come before it in the tour. Thus, a cost matrix for these quantities cannot express the value of all possible tours as can be done with Euclidean distances. Therefore, a more general expression for this problem is needed. The general optimization problem is given by the following.

Problem 2.3.3 *Consider the queue at time t where t is the time service is completed for a customer,*

- *Let $Q(t) = \{q_1, q_2, \ldots, q_n\}$ denote the queue at time t.*
- *Let $\mathcal{V}$ denote the tour on $Q(t)$.*
- *Let κ^s denote the switching time.*
- *Let $W(t, \kappa^s, \mathcal{V})$ denote the cost of tour $\mathcal{V}$.*

This leads to the problem formulation given by:

$$\textit{minimize} \quad W(t, \kappa^s, \mathcal{V}). \tag{2.17}$$

Additionally, the lasercom problem has a graph theoretic analog. However, due to the nature of the problem, the graph is non-euclidean and consequently cannot be depicted. The graph theoretic description of the lasercom system.

Problem 2.3.4 (Graph Theoretic) *Let be a dynamic graph $\mathcal{G}(t) = (V(t), E(t), W(t, \mathcal{V}))$, let $C(t) \subset V(t)$ be a set of vertices representing $N(t)$ stationary laser communication terminals on earth at time t, and let $\mathcal{V}$ be a tour on the vertices $C(t)$. Let S represent a single laser communication terminal in geosynchronous orbit. Determine the tour $\mathcal{V}$ of the vertices $C(t)$ that minimizes the sum of the edge weights $W(t, \mathcal{V})$.*

Next we will discuss what the cost matrices C and W should represent along with long term measures of performance.

2.4 Performance Measures

Now that a queueing system has been specified we will look at commonly used measures of the performance and effectiveness of the service disciplines used. The most common measures that are applicable to the lasercom problem are the *waiting time*, the *number of customers*, and the *system time*. These are all random variables and we will be seeking a probabilistic description of their behavior. In addition, while these measures are commonly used for traditional queueing systems, we will look at related measures for the lasercom system such as bandwidth and waiting time variance.

First we will look at waiting time. Let w_n^s denote the time from the arrival of the customer in the queue, till the time that the customer is chosen for service. Let κ^s denote the time to switch to the chosen user after that user is chosen for service. We will define the wait time w_n for each customer as the difference in time from when service starts for a customer and the time that the customer arrived in the queue. Therefore, the waiting time for an individual customer is given by $w_n = \kappa^s + w_n^s$. More specifically, we are interested in the average waiting time for all customers in the system.

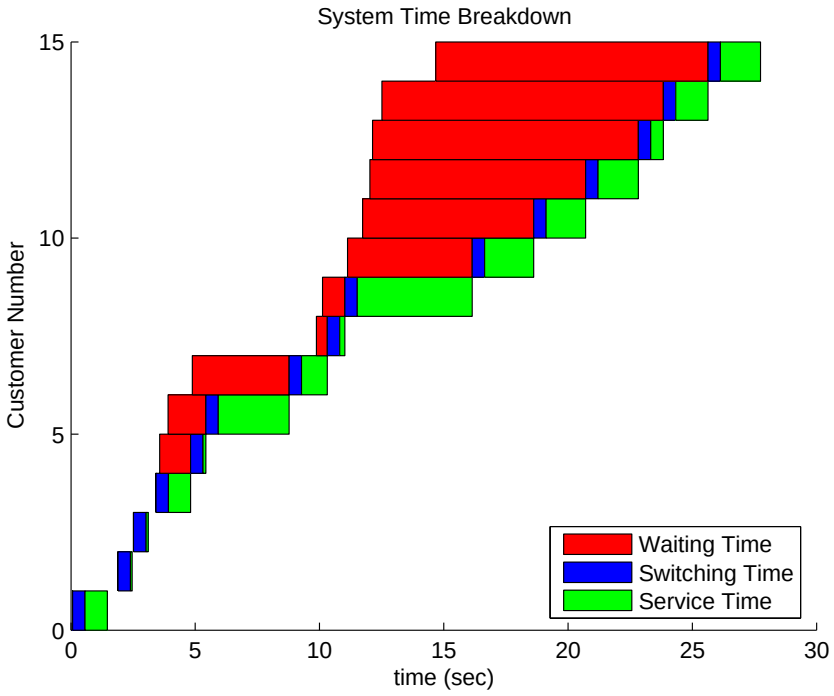


Figure 2.1: System time breakdown for first come first served service rule

Definition 2.4.1 (Average Waiting Time) *The average waiting time is given by:*

$$\overline{W} = \lim_{n\to\infty} w_n = \lim_{n\to\infty} \kappa^s + w_n^s = \kappa^s + \lim_{n\to\infty} w_n^s. \tag{2.18}$$

Another useful measure of system performance is the system time. This is the total time that a customer spends in the system. Let s_n denote the service time for each customer. The system time is $t_n = \kappa^s + w_n^s + s_n$ which leads to the average system time.

Definition 2.4.2 (Average System Time) *The average system time is given by:*

$$\overline{T} = \lim_{n\to\infty} t_n = \kappa^s + \lim_{n\to\infty} w_n^s + s_n. \tag{2.19}$$

Another extremely useful performance measure for this system is the system bandwidth. Let d_n denote the amount of data that each customer transmits. This data size is given by $d_n = rs_n$ where r is the connection speed for the link. Let f_n denote the time that the n^{th} customer finishes transmitting their data and let v_n denote the time that the queue is empty which is given by the equation:

$$v_n = \begin{cases} \tau_{n+1} - f_n & \tau_{n+1} > f_n, \\ 0 & \tau_{n+1} \leq f_n. \end{cases} \tag{2.20}$$

We will calculate the bandwidth for each customer in the time interval from the completion of service for the prior customer and the completion of service for the current customer. The bandwidth for the n^{th} customer is given by:

$$b_n = \frac{rs_n}{v_n + \kappa^s + s_n}. \tag{2.21}$$

Additionally, we are interested in the average bandwidth for the system.

Definition 2.4.3 (Average Bandwidth) *The average bandwidth is given by:*

$$\overline{b} \triangleq \left[\frac{\sum_n rs_n}{\sum_n t_n}\right] = \left[\frac{\sum_n rs_n}{\sum_n \kappa^s + v_n + s_n}\right]. \tag{2.22}$$

As we will show in section 3.5, for a stable system with constant switching time, the bandwidth is only a function of the traffic intensity parameter ρ. The two measures that we will try to minimize are the wait time and the variance of the wait time.

To define the wait time and variance of the wait time we will need to introduce some notation. At the time that each customer finishes service, let the current queue of size n where $n \in \mathbb{N}^+$ be denoted by $Q = \{q_1, q_2, \ldots, q_n\}$. Let the time that the last customer leaves the queue be denoted f_0 and the times that each customer in the queue finishes service for a given service order be denoted by f_i where $i = \{1, 2, \ldots, n\}$. For these n customers, let the order that they will be serviced be given by $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ where $\mathcal{V} \subseteq Q$ and $\mathcal{V} \cup Q = Q$ and $\mathcal{V} \cap Q = \emptyset$. Finally, let $\mathbf{s}_i$ be the service time required by each customer and κ^s the time to switch between two users.

The average wait time is given by the equation:

$$\overline{W} = \frac{\sum_{i=1}^{n}[(n+1) - i](\mathbf{s}_{v_i} + \kappa^s) + (f_0 - \tau_i)}{n}. \tag{2.23}$$

Next, let w_n denote the wait time for the n^{th} customer in the queue. This leads to the equation for the variance of the wait time given by:

$$\sigma_W^2 = \frac{\sum_{i=1}^{n}(w_i - \overline{W})^2}{n}. \tag{2.24}$$

The last two performance measure considered are priority and urgency. These measures could be used separately or in conjunction with the measures above to determine the order of service. These metrics are subjective to the actual system in service, but in general, priority will be a measure of the importance of the information being sent, while urgency will be a measure of how time critical the information is. Urgency will also come into play in the situations where data loss will occur because of buffer overruns in the customer. One method of implementing these metrics is as a scaling factor on the cost for that customer. Another method of implementation, is to allow customers to skip forward in line after the order of service has been decided, based on their priority and urgency.

Next we will develop two service rules based on average wait time and wait time variance. The first service rule will attempt to minimize the average wait time and the wait time variance. The second rule will minimize the average wait time at the expense of the wait time variance. Both service rules decide the next customer served based on the current knowledge of the customers in the queue at the finish of service of the previous customer. This problem is analogous to the traveling salesman problem. Namely, find an order of service that minimizes some metric. However, for the laser communications problem, the cost to 'travel' from one customer to the next varies with the route previously taken, whereas in the traveling salesman problem, the cost to travel from one customer to the next is independent of the route. This means that there is no simple graphical representation of the laser communications problem and the cost of all possible routes cannot be captured in matrix form. The consequence of the dependence of each customer's waiting time on the prior customers in the route, is that the cost of placing a customer at a particular spot in the service order is dependent on all the permutations of those prior customers. The problem then becomes intractable for large queue sizes, just as the traveling salesman problem becomes intractable for large numbers of cities. To reduce the size of the problem to polynomial time some assumptions and simplifications must be made.

2.4.1 Variation of Wait Time (VW) Service Rule

The first service rule that we will consider minimizes a combination of the average wait time and the variance of the wait time. Together these two quantities combine to form a balanced quality of service. Minimizing the average wait time will ensure quick service in general, while minimizing the variance will ensure that individual customers don't wait for excessively long periods of time to receive service.

To reduce the size of this optimization to a manageable level we will only determine the next customer to be served instead of attempting to determine the optimal order of service for the entire queue. We will do this by determining a relative cost for each customer in the queue to be served next. This cost vector is given by the equation,

$$w_i = (f_0 - \tau_i - \overline{W})^2 + \sum_{j \in \mathcal{V} \setminus v_j} \left[(f_0 - \tau_{v_j}) + (n - j + 1)(\kappa^s + s_{v_j}) - \overline{W} \right]^2. \tag{2.25}$$

Essentially, this is the variance of the wait time when each member of the queue in the first position with the remaining queue served in a first come first served manner. While this is not

the minimum variance, it allows for a reasonable choice of the next customer. Additionally, for a specific system traffic intensity, there is an expected value for the average waiting time. This service rule will attempt to minimize the individual waiting times to the average wait time instead of minimizing the wait time to zero which will be impossible.

Finally, we will use an estimate of the average wait time $\overline{W}$. While this can be calculated analytically for simple systems with a FCFS service rule, for a real system, an estimate will be more accurate than a theoretical value for $\overline{W}$ which relies on a first come first served service rule and assumes static traffic intensity. We will calculate the average wait time online by the formula,

$$\overline{W}_{n+1} = \frac{w_{n+1} - \overline{W}_n}{n+1}, \tag{2.26}$$

where $\overline{W}_0 = 0$. Additionally, for systems with quickly changing parameters, only the last m wait times would be used to calculate the average from wait time. This would allow the average wait time to more rapidly adjust to changing system parameters.

2.4.2 Nearest Insertion Service Rule

Next we will present a service rule that makes use of the Extended Lin-Kernighan heuristic. This service rule has the advantage of using all information available to minimize the cost. This heuristic will be developed in detail in Section 4.3. This heuristic is an iterative technique to find locally optimal solutions to the traveling salesman problem. While the algorithm was developed for a the traveling salesman problem with a static cost matrix that is invariant to the tour, it can be easily applied to the dynamic cost of the laser communications system. Essentially, the algorithm will remove segments of the tour and inserts them in the nearest position that minimizes a cost.

For the laser communications system we will minimize the average wait time given by (2.23), where this service rule and metric will be denoted by NI. Additionally, we will minimize the variance of the wait time (2.24) which will be denoted by NI2. The VW discipline is a simplified version of the NI2 discipline, and will be considered because of the decreased time to run the VW discipline.

Chapter 3

Analysis of Queue Behavior

3.1 Introduction

In this chapter, we will look at some theoretical results about the behavior of the laser communications queueing system. Specifically, we will determine how the bandwidth is related to the systems parameters, investigate the stability conditions for the system, derive equations that describe the average behavior of the wait time and the variance of the wait time of the system with non-zero wait time, and compare these results to the previously derived bounds for the problem.

3.2 Mathematical Preliminaries

One of the most important quantities for this system is the average number of users in the queue. This quantity is invariant to the arrival and service distribution.

Theorem 3.2.1 (Little's Theorem) *[27] If the limit* $\lambda = \lim_{t\to\infty} \lambda_t$ *and* $\overline{T}$ *do exist, then the limit* $\overline{N} = \lim_{t\to\infty} \overline{N}_t$ *does exits, too, and the relation*

$$\overline{N} = \lambda \overline{T} \tag{3.1}$$

holds, where $\overline{T}$ *is the average system time and* $\overline{N}$ *is the average number of customers in the systems.*

The proof can be found in [27]. For completeness we give the proof here. **Proof.** We introduce the notation $\delta(t)$ for the number of departures from the system during $[0, t]$ and $N(t)$ for the number of users in the system. If the system starts empty, then these definitions imply the relation

$$N(t) = \alpha(t) - \delta(t) \tag{3.2}$$

for all times t. The total time that all users have spent in the system during $[0, t]$ is given by

$$\gamma(t) \triangleq \int_0^t N(s)\mathrm{ds}. \tag{3.3}$$

If we define T_t as the system time per user averaged over all users in the interval $[0, t]$, then the definitions of $\alpha(t)$ and $\gamma(t)$ imply the relation

$$T_t = \frac{\gamma(t)}{\alpha(t)}. \tag{3.4}$$

The average number of users in the system during $[0, t]$ can be obtained as

$$\overline{N}_t = \frac{\gamma(t)}{t} = \frac{\gamma(t)}{\alpha(t)} \cdot \frac{\alpha(t)}{t} = \lambda_t T_t, \tag{3.5}$$

where the last equality comes from (3.4) and the definition of λ_t. If the limits λ and $T = \lim_{t\to\infty} T_t$ exist, then the stated relation $\overline{N} = \lambda T$ follows for t tending to infinity. □

A consequence of Little's result is:

$$\overline{N}_q = \lambda \overline{W}, \tag{3.6}$$

where $\overline{N}_q$ is the average number of customers in the queue and $\overline{W}$ is the average waiting time.

For a statistical process, we can estimate the mean $\eta(t)$ with their *ensemble average*

$$\eta(t) \simeq \frac{1}{n} \sum_{i=1}^{n} x(t, \zeta_i), \tag{3.7}$$

where $x(t, \zeta_i)$ are samples of the process. A *time-average* for a single sample $x(t, \zeta)$ of $x(t)$ is given by:

$$\overline{x} = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t, \zeta) dt. \tag{3.8}$$

Definition 3.2.1 *[25] A stochastic process $x(t)$ is called* ergodic *if its ensemble averages equal appropriate time averages.*

This means that with a probability of 1, any statistic of $x(t)$ can be determined from a single sample $x(t, \zeta)$. Additionally, ergodic systems "forget" their initial state, that is, the probability distribution converges to a limiting distribution regardless of initial condition [27].

3.3 Maximum Connection Time

An additional consideration is the length of time that the satellite allows the connection to remain open. We will show in the following that the optimum bandwidth occurs when each terminal finishes sending all data before the next terminal receives service. This is feasible for the laser communications system because of the high transfer rates. For implementation however, an upper bound on the connection time would need to be set to prevent the system from being dominated by any one user.

Lemma 3.3.1 *Let $\mathcal{T}$ be a set of n earth terminals with a static amount of data d_i, $i = 1, \ldots, n$ to be transferred. Let S be a satellite in geosynchronous orbit with connection bandwidths b_i, $i = 1, \ldots, n$ to each of the terminals in $\mathcal{T}$. Given a constant switching time between terminals $t_s > 0$, the minimum average bandwidth $\bar{b}_A$ occurs when each terminal transmits all data to the satellite before the satellite switches to the next terminal.*

Proof. If the satellite transfers all the data from each terminal before switching to the next, the average bandwidth for the system is given by:

$$b_A = \frac{\sum_{i=1}^{n} d_i}{n t_s + \sum_{i=1}^{n} t_i}. \tag{3.9}$$

Assume the satellite takes two connections to receive all the data from one terminal. This splits one d_i into two $d_i = d_{i_1} + d_{i_2}$. The time to transfer the data t_i becomes

$$t_i = \frac{d_i}{b_i} = \frac{d_{i_1} + d_{i_2}}{b_i} = \frac{d_{i_1}}{b_i} + \frac{d_{i_2}}{b_i} = t_{i_1} + t_{i_2}. \tag{3.10}$$

With an index change the set of data to be transferred becomes $d_i \in \mathbb{Z}^n \to \hat{d}_j \in \mathbb{Z}^{n+1}$ and the set of transfer times becomes $t_i \in \mathbb{Z}^n \to \hat{t}_j \in \mathbb{Z}^{n+1}$. The average bandwidth for this system is given by:

$$\hat{b}_A = \frac{\sum_{j=1}^{n+1} \hat{d}_j}{(n+1)t_s + \sum_{j=1}^{n+1} \hat{t}_j}. \tag{3.11}$$

Since,

$$\sum_{i=1}^{n} d_i = \sum_{i=1}^{n+1} \hat{d}_j, \tag{3.12}$$

and

$$\sum_{i=1}^{n} t_i = \sum_{i=1}^{n+1} \hat{t}_j, \tag{3.13}$$

substituting (3.12) and (3.13) into (3.11) yields

$$\hat{b}_A = \frac{\sum_{i=1}^{n} d_i}{(n+1)t_s + \sum_{i=1}^{n} t_i} < b_A. \tag{3.14}$$

□

Remark 3.3.1 *This result assumes that all terminals are available for connection by the satellite.*

Remark 3.3.2 *For a system with constant switching time, if all data is removed from each terminal before switching, the order that the terminals are visited does not affect the bandwidth, which, in fact, is the minimum possible bandwidth. For systems with small switching times compared to the time to transmit data, the effect on the average bandwidth is small. In systems such as laser communications where the ratio of the switching time and the average time to transmit a set of data is high, the effect on the average bandwidth is very large which makes small packet sizes in the network highly inefficient.*

Since the order that the terminals are selected in this system has no effect on average bandwidth, other criterion will be used in selecting the order of service, such as wait time and terminal priority. For a laser communication system with moving ground targets, the switching time becomes a function of the wait time for an individual terminal. The longer the terminal waits for service, the further it has moved from its original location which leads to a larger switching time due to the increased uncertainty in the terminal's location.

We will require the satellite to finish service for a customer before switching to optimize the bandwidth. This assumption is valid for laser communications satellites because the high bandwidth means that an entire transmission takes a relatively short amount of time. However, this means that the customers must store data until they are ready to transmit which requires a certain amount of on-board memory. Additionally, customers that are continually generating data will have to decide how often to burst a transmission to the satellite. This becomes a tradeoff between bandwidth and how critical it is to transmit the data immediately.

3.4 Birth-Death Queueing System in Equilibrium

The classic M/M/1 queueing system can be thought of in terms of a birth death process. From this perspective, the size of the queue is the variable of interest. A *birth* is when a customer arrives in the queue and a *death* is when that customer departs. These arrival and departure rates are again based on Poisson processes. An interesting question is what the equilibrium behavior of this system is (assuming the system is stable).

3.4.1 General Solution

To study the asymptotic behavior of this queueing system we will treat the queue as a chain where the system can move only from one link to the adjacent link with a probability governed by a probability distribution. The number of the link is the number of customers in the queue.

The following is based on [27]. Let E_k denote the state that there are k customers in the queue. The birth rate for state E_k is given by λ_k and the death rate for state E_k is given by μ_k. It is assumed that $\mu_0 = 0$ since there is only a nonnegative number of customers in the queue. For this system the transient response is given by the Kolomogorov forward equations of motion:

$$\frac{dP_k(t)}{dt} = -(\lambda_k + \mu_k)P_k(t) + \lambda_{k-1}P_{k-1}(t) + \mu_{k+1}P_{k+1}(t), \quad k \geq 1, \tag{3.15}$$

$$\frac{dP_0(t)}{dt} = -\lambda_0 P_0(t) + \mu_1 P_1(t), \quad k = 0. \tag{3.16}$$

Assuming that the limit of (3.19) exists, then setting the limit of $dP_k(t)/dt$ as $t \to \infty$ to zero yields the equilibrium equations

$$0 = -(\lambda_k + \mu_k)p_k + \lambda_{k-1}p_{k-1} + \mu_{k+1}p_{k+1}, \quad k \geq 1, \tag{3.17}$$

$$0 = -\lambda_0 p_0 + \mu_1 p_1, \tag{3.18}$$

where

$$p_k \triangleq \lim_{t\to\infty} P_k(t) \tag{3.19}$$

denote the limiting probability that the system is in state E_k. To remove the boundary condition given by equation 3.18 we will define the following:

$$\lambda_{-1} = \lambda_{-2} = \lambda_{-3} = \ldots = 0,$$
$$\mu_0 = \mu_{-1} = \mu_{-2} = \ldots = 0,$$
$$p_{-1} = p_{-2} = p_{-2} = \ldots = 0.$$

Thus for all values of k, (3.17) and (3.18) become:

$$\frac{dP_k(t)}{dt} = -(\lambda_k + \mu_k)P_k(t) + \lambda_{k-1}P_{k-1}(t) + \mu_{k+1}P_{k+1}(t), \tag{3.20}$$

for $k \in [-\infty, \infty]$. The following conservation property must also exist:

$$\sum_{k=0}^{\infty} p_k = 1. \tag{3.21}$$

The solution for p_k is given by:

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}. \tag{3.22}$$

By applying the conservation property (3.21) the initial limiting probability is given by:

$$p_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}}. \tag{3.23}$$

These are the principle equations for a basic queue.

To determine if a steady state solution to Eqs. 3.22 and 3.23 exist, we will require that $p_0 > 0$ so that those equation represent a probability distribution. This condition means that the system occasionally empties. The system is ergodic if the following conditions hold:

$$S_1 \triangleq \sum_{k=0}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} < \infty, \tag{3.24}$$

$$S_2 \triangleq \sum_{k=0}^{\infty} \left(\frac{1}{\lambda_k \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}} \right) = \infty. \tag{3.25}$$

If there exists some k_0 such that for all $k \geq k_0$ the term $\lambda_k/\mu_k < 1$ these conditions hold.

3.4.2 M/M/1 Queueing System

In the following we will consider the equilibrium behavior of the laser communications system. For this birth-death system, the arrival and departure coefficients are constant and are given by $\lambda_k = \lambda$ and $\mu_k = \mu \frac{\mu}{\mu + \kappa^s} = b\mu$. Inserting these coefficients into Eq. 3.22 we arrive at:

$$p_k = p_0 \left(\frac{\lambda}{b\mu} \right)^k, \quad k \geq 0. \tag{3.26}$$

Therefore, for the system to be ergodic

$$S_1 = \sum_{k=0}^{\infty} \frac{p_k}{p_0} = \sum_{k=0}^{\infty} \left(\frac{\lambda}{b\mu} \right)^k < \infty, \tag{3.27}$$

$$S_2 = \sum_{k=0}^{\infty} \frac{1}{\lambda(p_k/p_0)} = \sum_{k=0}^{\infty} \frac{1}{\lambda} \left(\frac{b\mu}{\lambda} \right)^k = \infty. \tag{3.28}$$

The condition (3.28) is satisfied when $\lambda/\mu \leq b$, so the condition $\rho < \frac{\mu}{\mu + \kappa^s}$ is necessary and sufficient for ergodicity. Next, the limiting probability for an empty queue is given by:

$$p_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \left(\frac{\lambda}{b\mu} \right)^k}. \tag{3.29}$$

Since, $\lambda < b\mu$, this equation becomes,

$$p_0 = 1 - \frac{\lambda}{b\mu} = 1 - \frac{\rho}{b}. \tag{3.30}$$

The stability condition then becomes $0 \leq \rho < 1$ which also ensures that $p_0 > 0$. Finally, inserting (3.30) into (3.26) yields:

$$p_k = \frac{1}{b}(b - \rho)\left(\frac{\rho}{b}\right)^k. \tag{3.31}$$

This equation give the steady state probability of finding k customers in the queue.

3.5 Average Bandwidth

Of interest is the behavior of the systems average bandwidth over a range of switching times. Counter-intuitively, the service protocol used has no effect on the average bandwidth. The only factors that affect bandwidth are the arrival rate, the service time, and the switching time. When the system has low traffic intensity, on average, a customer does not have to wait for service. In this case, the average bandwidth is given by:

$$\bar{b}_L = \frac{rE[s_n]}{E[t_n]} = \frac{r(1/\mu)}{1/\lambda} = r\rho, \tag{3.32}$$

which is just the fraction of time between the arrival of customers that the server is busy.

When the traffic intensity is high, the server is busy constantly, thus the two parameters that affect the bandwidth are the switching time, and the average service time. For high traffic intensity, the average bandwidth is given by:

$$\bar{b}_H = \frac{rE[s_n]}{E[s_n] + \kappa_s} = \frac{r}{1 + \mu\kappa_s}. \tag{3.33}$$

As Figure 3.1 shows, there is a sharp transition between the low and high traffic bandwidths. This is a reflection of the system becoming unstable. As the switching time increases, the amount of the full system capacity that can be used decreases.

3.6 Stability Analysis

By adding a switching time to the queueing system, we reduce the maximum allowable traffic intensity because of the time 'wasted' by switching. By the assumptions for (3.33), the server is constantly in use. This is by definition, unstable, as the queue never empties. Therefore, the new condition for stability becomes:

$$\rho \leq \frac{\mu}{\mu + \kappa^s} = \frac{1}{1 + \kappa^s\mu} = b. \tag{3.34}$$

The quantity b represents the fraction of the total service time left over after the switching delay that can be used for service.

As Figure 3.2 shows, the range of traffic intensities that are stable quickly drops with high switch time compared to the service time. When the switching time and service times are equal, the system can handle only half its normal traffic, and at ten times the service time, less than 10%.

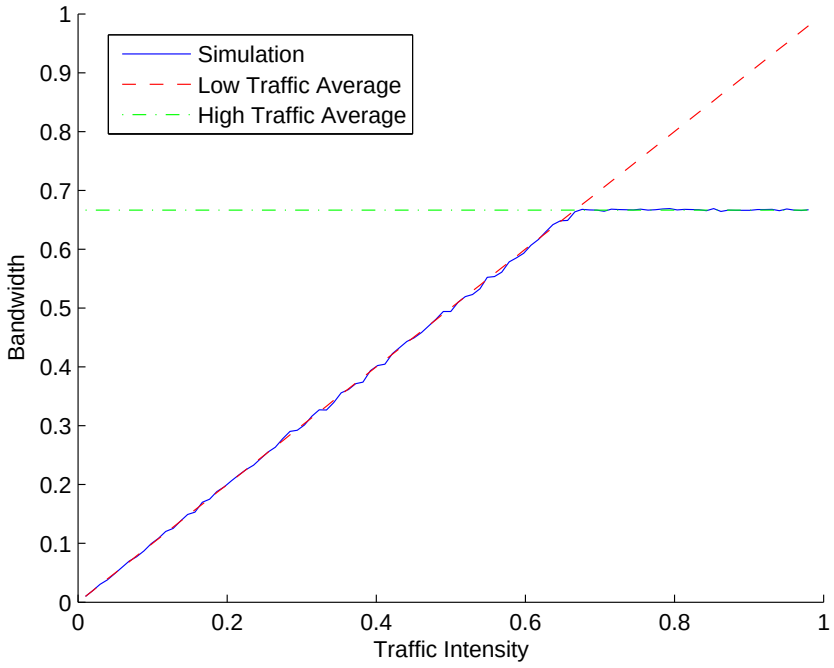


Figure 3.1: Average Bandwidth

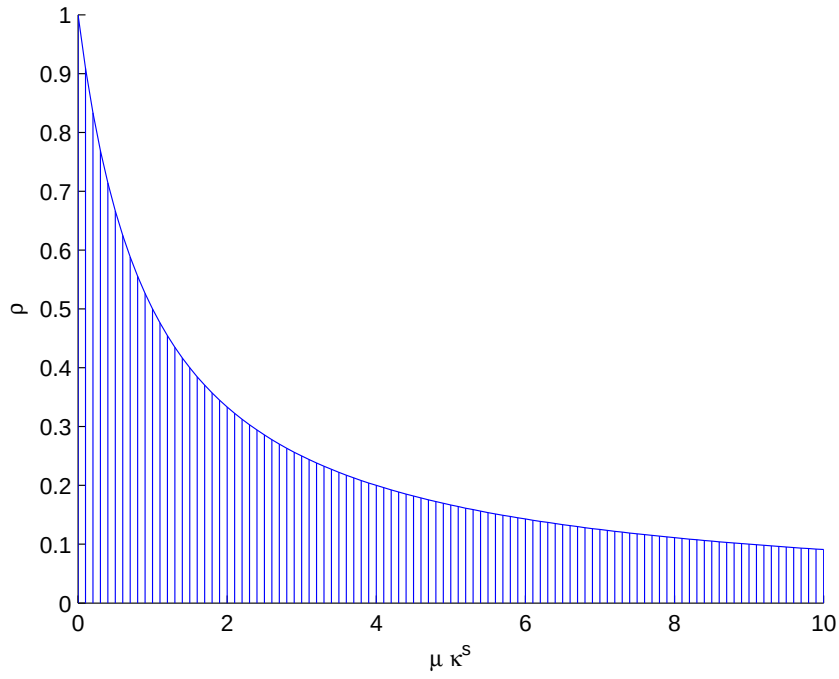


Figure 3.2: Stability region for varying switching time.

3.7 Wait Time

To determine the expected value of the waiting time we will apply Little's theorem:

$$\overline{N}_q = \lambda \overline{T} = \lambda(\overline{W}). \tag{3.35}$$

Solving for the average waiting time yields:

$$\overline{W} \geq \frac{\overline{N}_q}{\lambda}. \tag{3.36}$$

The average number of customers in the system and is given by the equation

$$\begin{aligned}
E[N_q] = \overline{N}_q &\leq \sum_{k=0}^{\infty} k p_k, \\
&= (1 - \frac{\rho}{b}) \sum_{k=0}^{\infty} k \left(\frac{\rho}{b}\right)^k, \\
&= (1 - \frac{\rho}{b}) \rho \frac{\partial}{\partial \rho} \sum_{k=0}^{\infty} \left(\frac{\rho}{b}\right)^k, \\
&= (1 - \frac{\rho}{b}) \rho \frac{\partial}{\partial \rho} \left(1 - \frac{\rho}{b}\right)^{-1}, \\
&= (1 - \frac{\rho}{b}) \rho \frac{1}{b} \left(1 - \frac{\rho}{b}\right)^{-2}, \\
&= \frac{\rho}{b\left(1 - \frac{\rho}{b}\right)}, \\
&= \frac{\rho}{b - \rho}, \\
&= \frac{\rho}{\frac{\mu}{\mu + \kappa^s} - \rho}.
\end{aligned} \tag{3.37}$$

This leads to the expression for the average waiting time given by:

$$\overline{W} = \frac{\rho}{\lambda(b - \rho)}. \tag{3.38}$$

We can see in Figures 3.3 and 3.4 that the expected values for the average queue size and the average wait time closely follow the simulated values for a large range of switching times.

Figure 3.5 shows that the when the switching time is zero, the new expression for the expected value of the wait time and the Bertsimas and Van Ryzin bound are identical. However, when the switching time is increased, the Bertsimas and Van Ryzin bound does significantly worse.

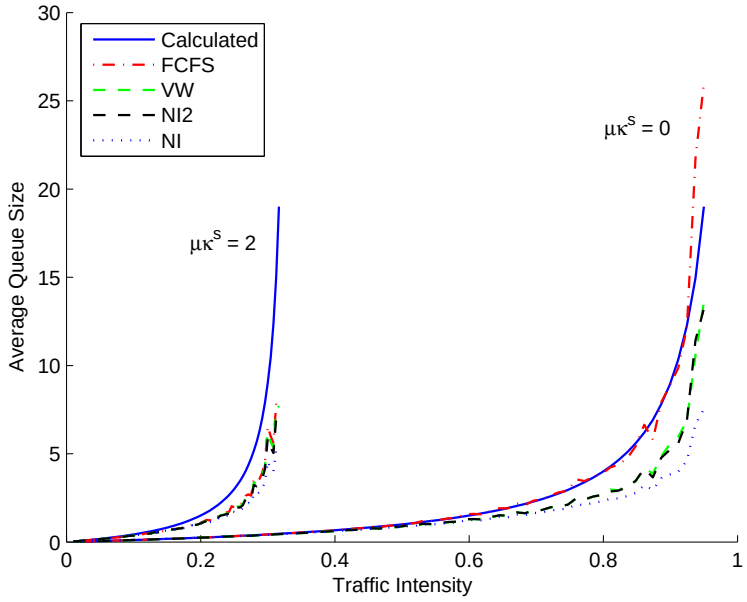


Figure 3.3: Comparison of the calculated and simulated average queue size for various service disciplines.

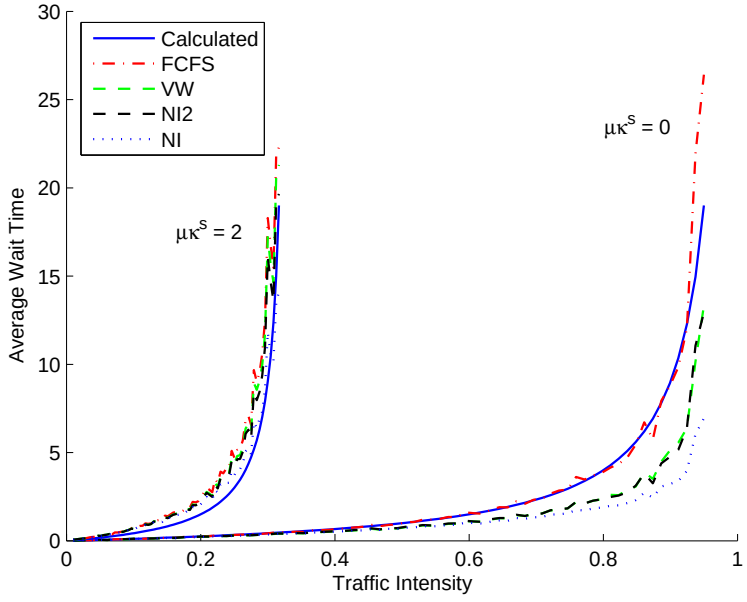


Figure 3.4: Comparison of the calculated and simulated average wait time for various service disciplines.

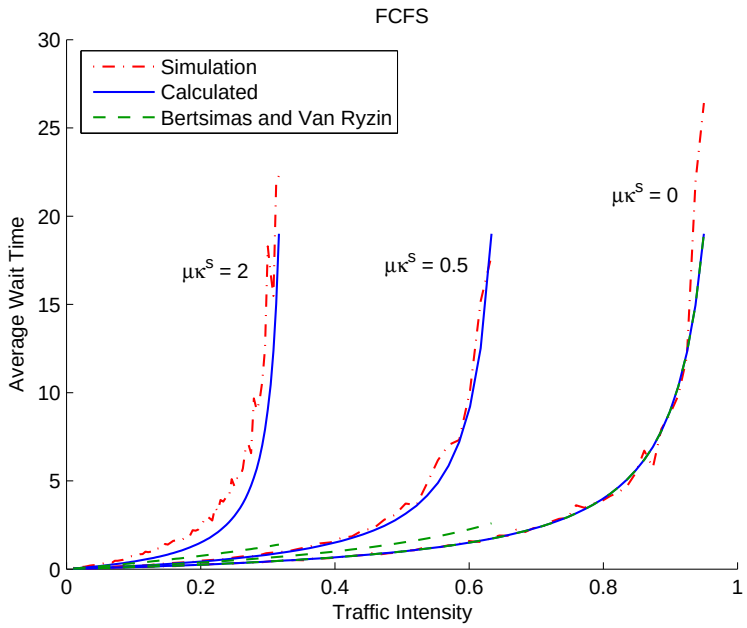


Figure 3.5: Comparison of the new expected value for the average wait time and the Bertsimas and Van Ryzin bounds.

3.8 Wait Time Variance

Similarly we can calculate the variance of the number of customers in the system with switching time:

$$\begin{aligned}
\sigma_N^2 &= \sum_{k=0}^{\infty}(k-\overline{N})^2 p_k, \\
&= \sum_{k=0}^{\infty} k^2 p_k + \overline{N}^2 \sum_{k=0}^{\infty} p_k - 2\overline{N}\sum_{k=0}^{\infty} k p_k, \\
&= \sum_{k=0}^{\infty} k^2 p_k + \overline{N}^2 - 2\overline{N}^2, \\
&= \left(1-\frac{\rho}{b}\right)\sum_{k=0}^{\infty} k^2 \left(\frac{\rho}{b}\right)^k - \overline{N}^2, \\
&= \frac{\frac{\rho}{b}\left(1+\frac{\rho}{b}\right)}{\left(1-\frac{\rho}{b}\right)^2} - \frac{\rho^2}{b^2\left(1-\frac{\rho}{b}\right)^2}, \\
&= \frac{b\rho}{(b-\rho)^2}.
\end{aligned} \tag{3.39}$$

The variance of the wait time then becomes:

$$\begin{aligned}
E[(W-\overline{W})^2] &= E[W^2] - (E[W])^2, \\
&= E[N^2S^2] - \frac{\rho^2}{\mu^2(1-\rho)^2}, \\
&= (\overline{N^2})(\overline{S^2}) - \frac{\rho^2}{\mu^2(b-\rho)^2}, \\
&= (\sigma_N^2 + (\overline{N})^2)(\sigma_S^2 + (\overline{S})^2) - \frac{\rho^2}{\mu^2(1-\rho)^2}, \\
&= \left(\frac{b\rho+\rho^2}{(b-\rho)^2}\right)\left(\frac{2}{\mu^2}\right) - \frac{\rho^2}{\mu^2(b-\rho)^2}, \\
&= \frac{2b\rho+\rho^2}{\mu^2(b-\rho)^2}.
\end{aligned} \tag{3.40}$$

Figure 3.6 shows that the simulated wait time variance for a range of service disciplines and switching times is close to the expected value for the wait time variance.

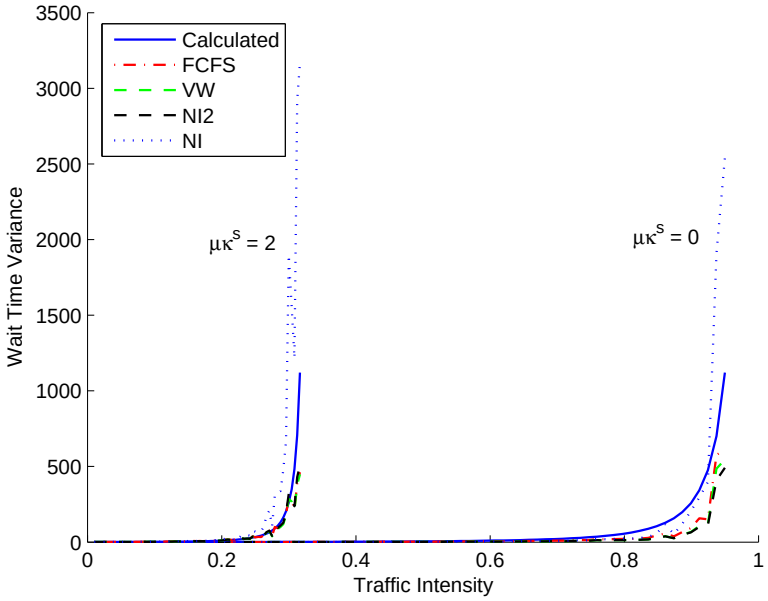


Figure 3.6: Comparison of the calculated and simulated wait time variance for various service disciplines.

Chapter 4

Performance Analysis

In this chapter we will present the simulation results for the laser communications system. Specifically, in Section 4.1 we will look at the solution convergence and determine the time interval required for convergence. Section 4.2 will compare the service rules developed in Section 2.4 to the two most commonly used service rules. Finally, Section 4.3, will discuss in detail the nearest insertion algorithm and how it can be applied to the laser communications problems.

4.1 Convergence of Solutions

For a stable queueing system, the average number of users in the queue will converge to a steady state value as time goes to infinity. When simulating the behavior of a queue, the time that the system takes to converge to within a small percentage of the steady state value, or *settling time*, is of interest. This will give confidence in the results, while allowing the simulation to complete in a reasonable amount of time. The author is not aware of any analytical results on this subject, therefore we will rely on simulations of a standard M/M/1 queue to determine a reasonable approximation for the settling time. The settling time for a queue is a function of the traffic intensity parameter ρ. As $\rho \to b$, the time to convergence increases.

We will define the low traffic region as the region where $\rho \to 0$. As shown by Figures 4.1 and 4.2, the convergence for the low traffic condition is safely around 15000 customers.

We will define the medium traffic region as having on average one customer in the queue. Figures 4.3 and 4.4 show that medium traffic queues converge in approximately 35000 to 50000 customers served.

High traffic regions will be defined as $\rho \to b$. Figures 4.5 and 4.6 show that the system with high traffic has large variations in its system parameters, even to 200000 customers served. It is obvious from the equation for the average number of customers in the queue, given by: $\overline{N} = \rho/(b-\rho)$, that this quantity goes asymptotically to infinity as $\rho \to b$. Thus the system is approaching instability as $\rho \to b$. As you approach this bound it becomes more likely that there will be periods of time when the customers are entering the system more quickly than they are leaving although on average $\rho < b$. So, intuitively, the more traffic in the system, the longer it will take for the system to settle to a steady state.

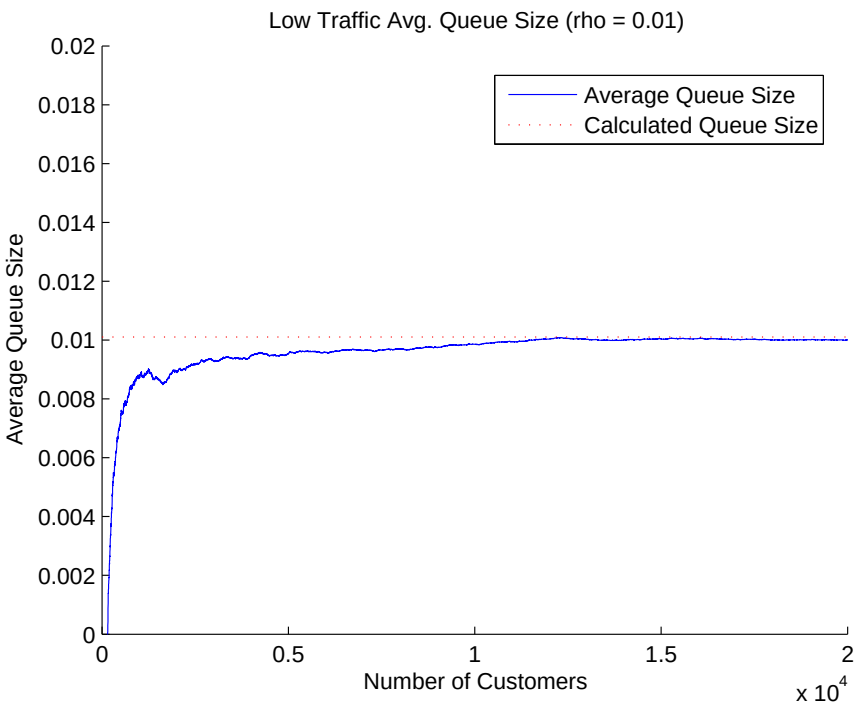


Figure 4.1: Low Traffic Average Queue.

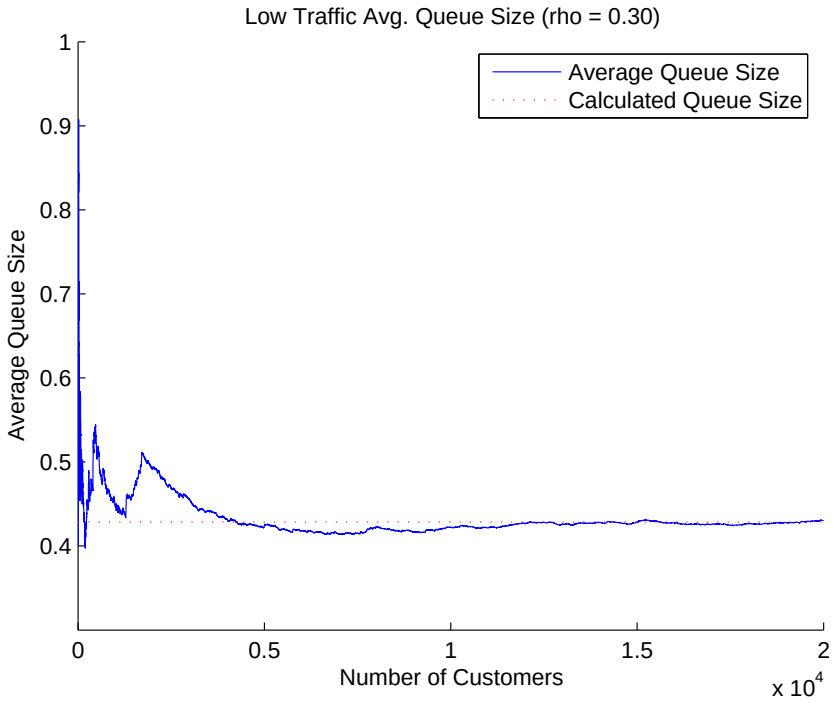


Figure 4.2: Low Traffic Average Queue.

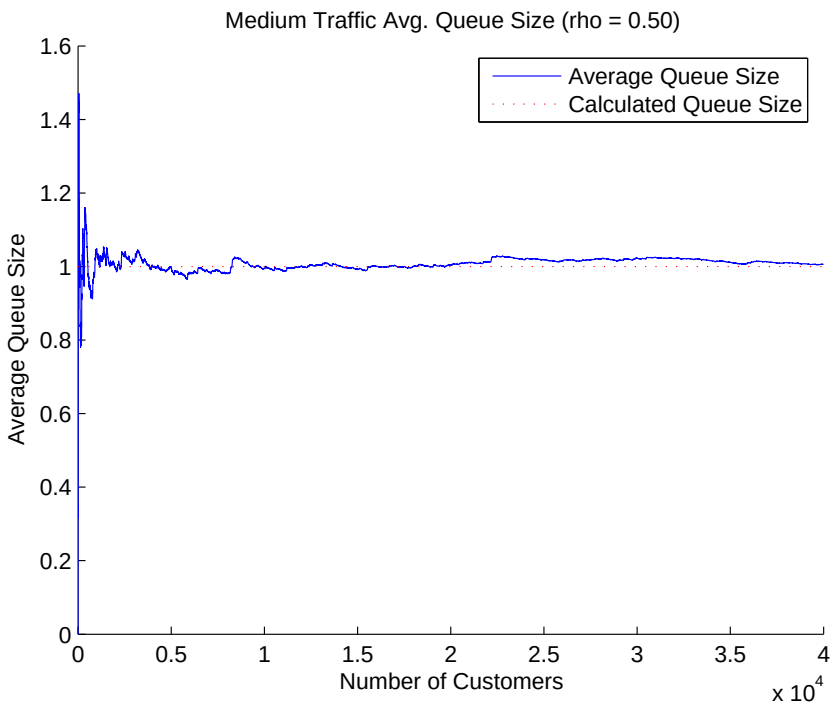


Figure 4.3: Medium Traffic Average Queue.

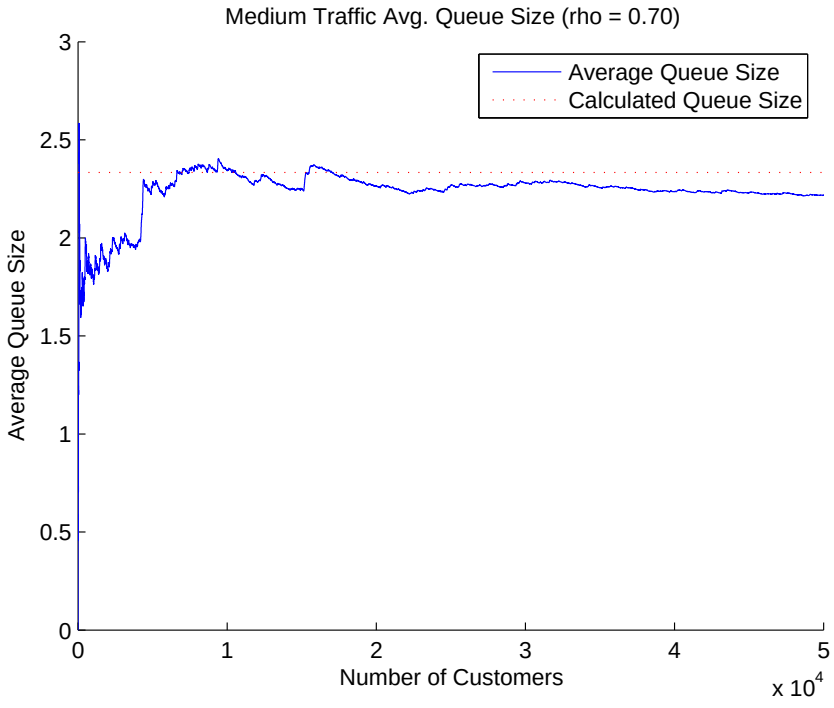


Figure 4.4: Medium Traffic Average Queue.



Figure 4.5: Heavy Traffic Average Queue.

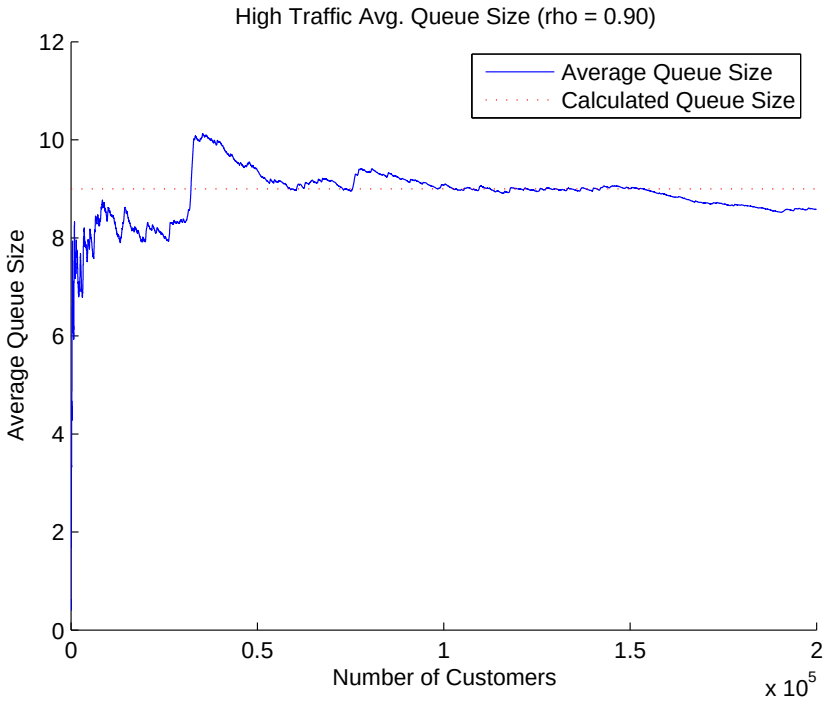


Figure 4.6: Heavy Traffic Average Queue.

4.2 Stationary Target Service Rules

In the simplified case under consideration where the customers are stationary with constant switching time we are interested in which service rules minimize the wait time and the variance of the wait time. We will investigate five service disciplines and their performance over a range of conditions for the laser communication system.

The first protocol is the ubiquitous first come first served (FCFS) rule also known as first in first out (FIFO). This service discipline is used the vast majority of the time in application and analysis. The second protocol is last come last served (LCLS). While not as common, this service discipline is used in some applications. Finally, we will test the variance of the wait (VW) service discipline given by (2.25) and the nearest insertion protocols NI and NI2 which minimize the average wait time and the variance of the wait time respectively.

The FCFS rule has been shown to be optimal for low traffic ($\rho \to 0$) situation. In fact, in low traffic, all service rules collapse to FCFS due to the fact that there is only one customer in the queue at a time. Thus, we will expect all service disciplines to approach the FCFS wait time and variance as $\rho \to 0$.

4.2.1 Zero Switching Time

To start with, we will look at the system in the ideal case, with zero switching time. This is the standard queuing system analyzed in books such as [27]. Figures 4.7 and 4.8 clearly show that the VW protocol outperforms both the FCFS and the LCFS protocols, especially in the high traffic regions of the system. As expected the NI2 protocol does slightly better than the VW protocol and the NI protocol has the most improvement.

Figure 4.9 shows that the LCFS discipline has large variation in the wait time as the system traffic intensity increases, while the FCFS and VW disciplines are comparable in the scale of variance. The NI2 protocol has slightly higher variance in exchange for a lower average wait time, while the NI protocol has the highest variance.

Figure 4.10 reveals that the VW discipline has a lower variance than the NI2 protocol for the entire range of traffic intensities, while the NI protocol does better than the FCFS until the traffic intensity becomes high.

4.2.2 Medium Switching Time

In this section we will simulate the system with a medium switching time. In this case, the switching time is 50% of the average service time μ. As figures 4.11 and 4.12 show similar results as the standard queueing system. The VW service discipline outperforms both the FCFS and the LCLS disciplines in average waiting time, especially in high traffic situations, while the NI2 and NI protocols outperform every other one.

As shown in Figure 4.13, the LCLS discipline show very poor performance in minimizing the variation as the system becomes heavily loaded. This is because under these conditions, a customer might get 'left behind'. Because the queue empties only rarely, the first person in the queue might wait for quite a while before being served, while another customer who just joined the queue might be served immediately. Figure 4.14 shows that the VW and NI2 protocols outperform the FCFS while the NI protocol is sacrifices variance for bandwidth in high traffic situations.

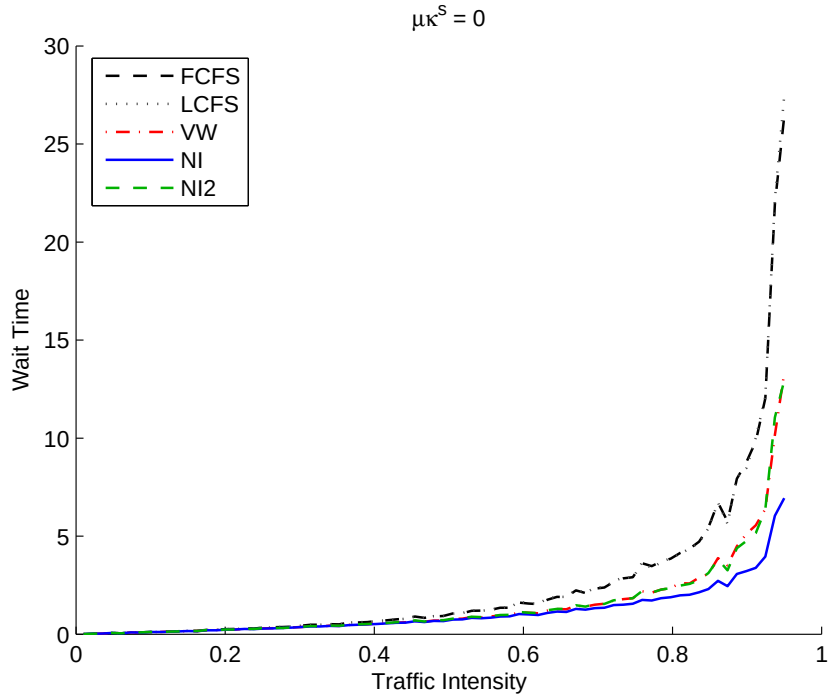


Figure 4.7: Average wait time for three service disciplines with zero switching time.

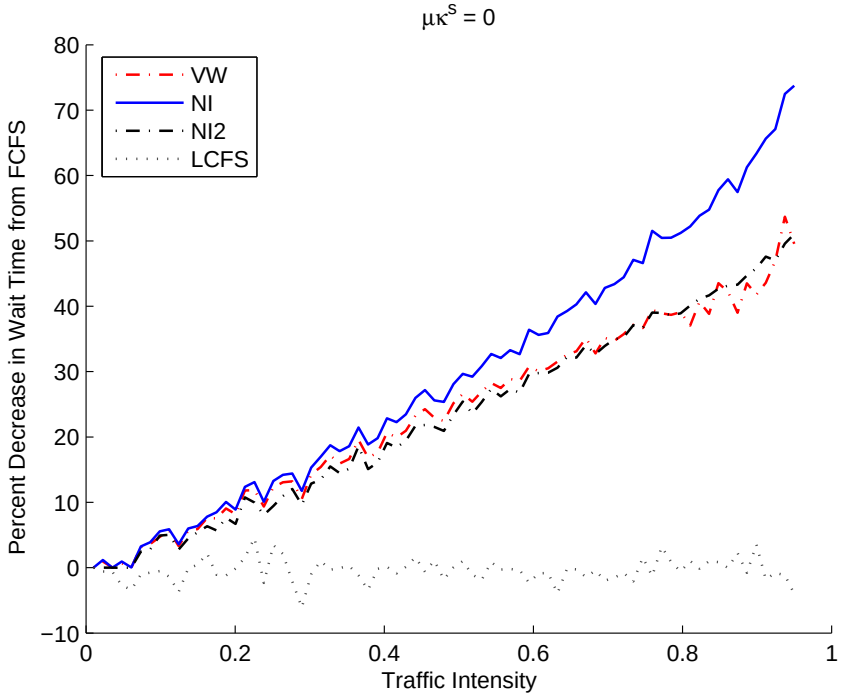


Figure 4.8: The percent decrease in the wait time from the first come first served service rule.

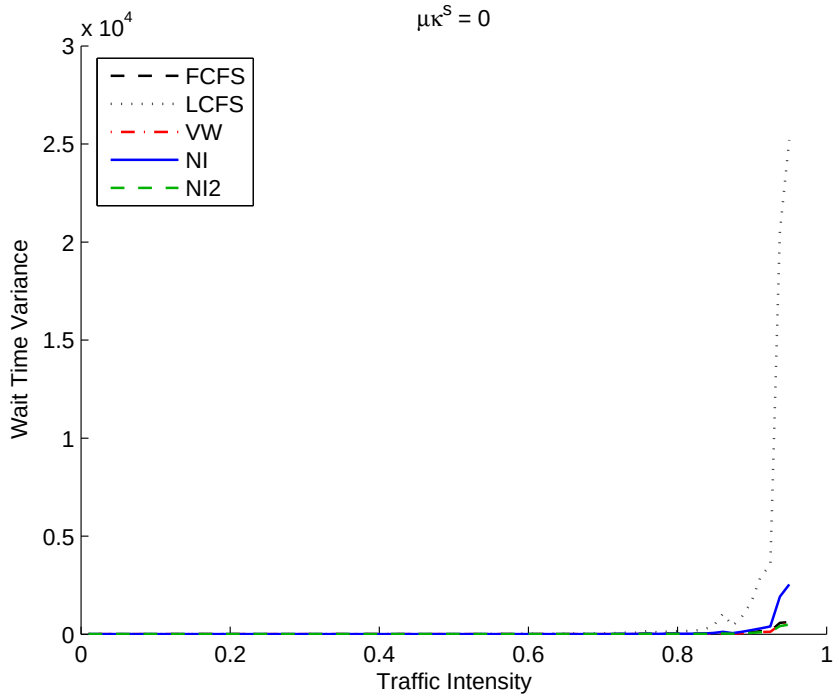


Figure 4.9: Variance of the wait time with zero switching time.

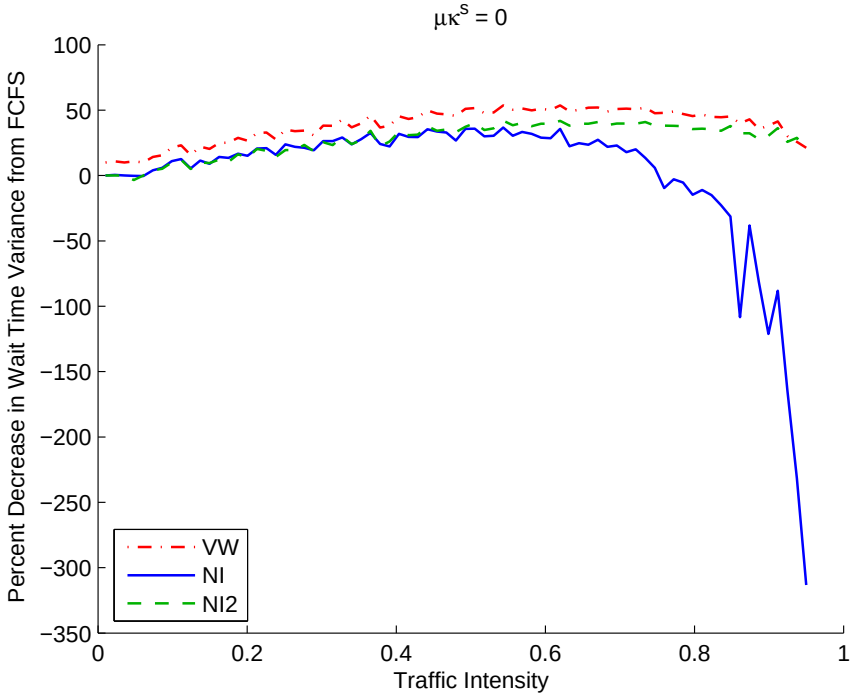


Figure 4.10: The percent decrease in wait time variation from the first come first served service rule.

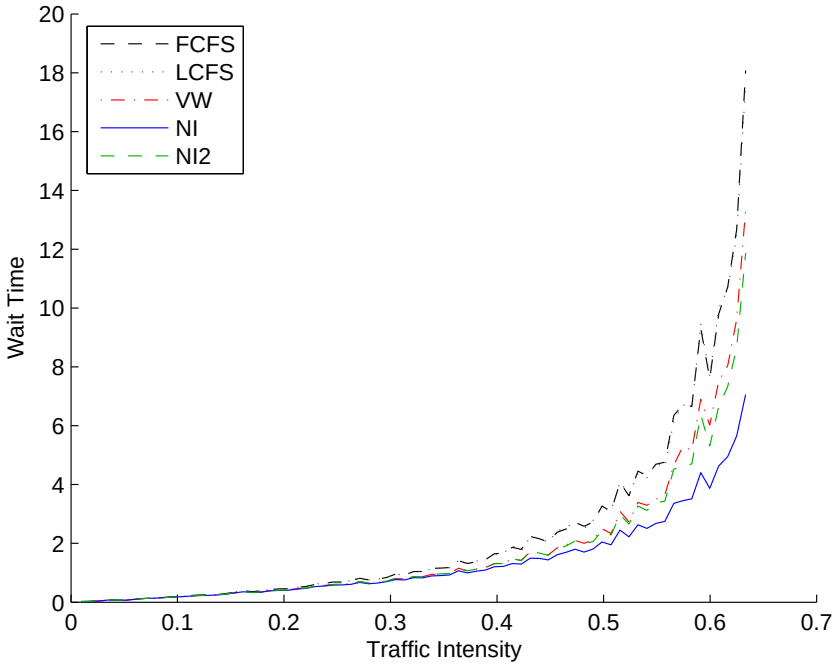


Figure 4.11: Average wait time for three service disciplines with medium switching time.

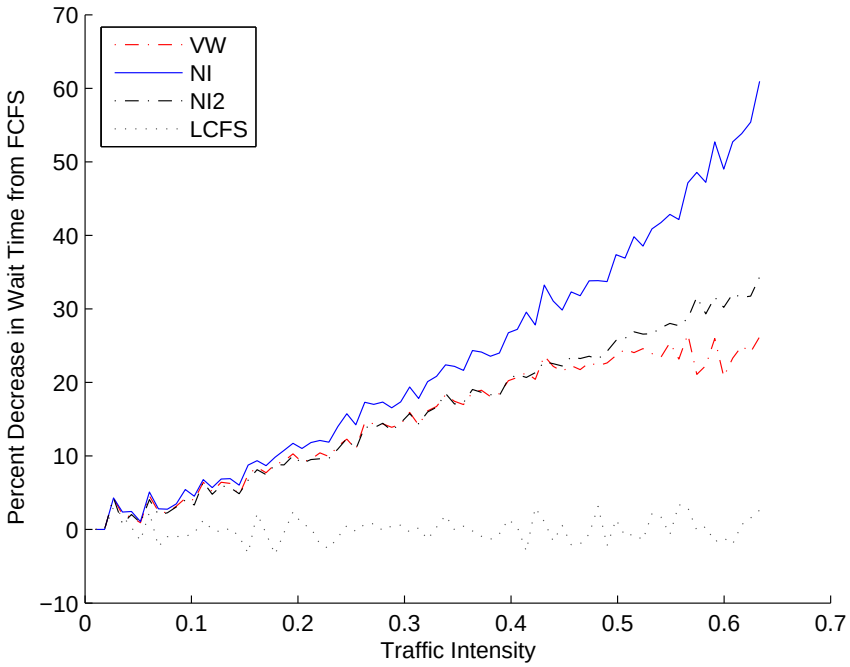


Figure 4.12: The percent decrease in the wait time from the first come first served service rule.

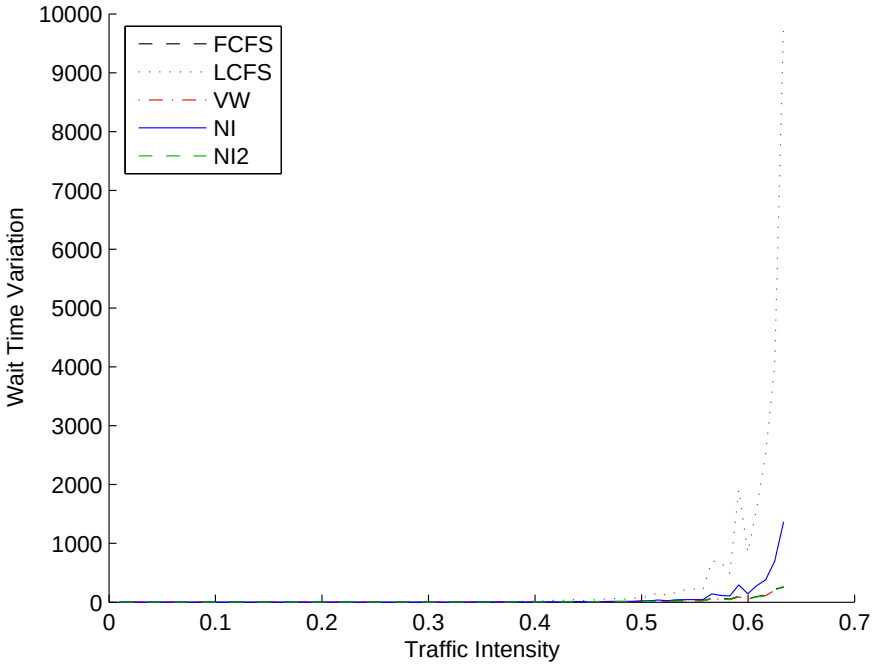


Figure 4.13: Variance of the wait time with medium switching time.

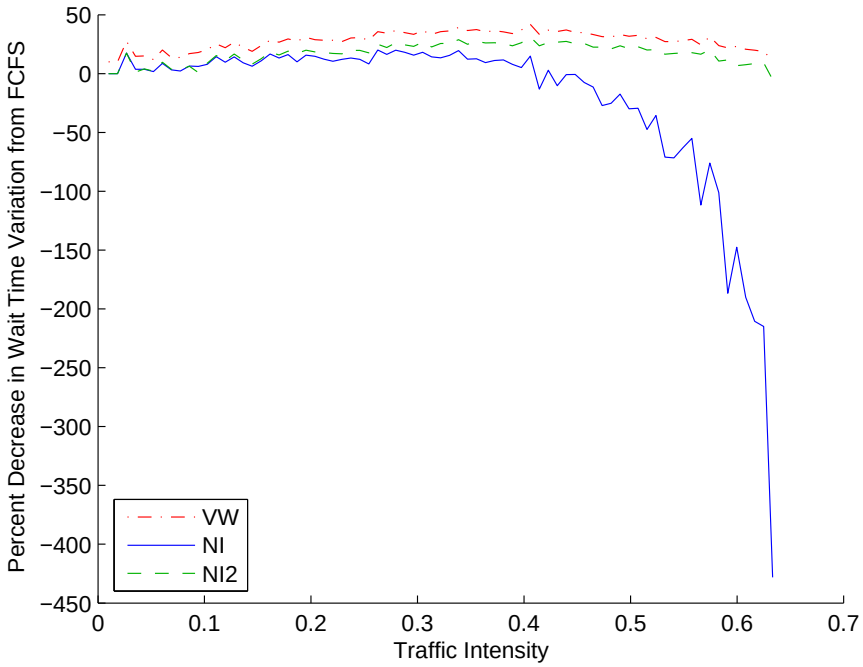


Figure 4.14: The percent decrease in wait time variation from the first come first served service rule.

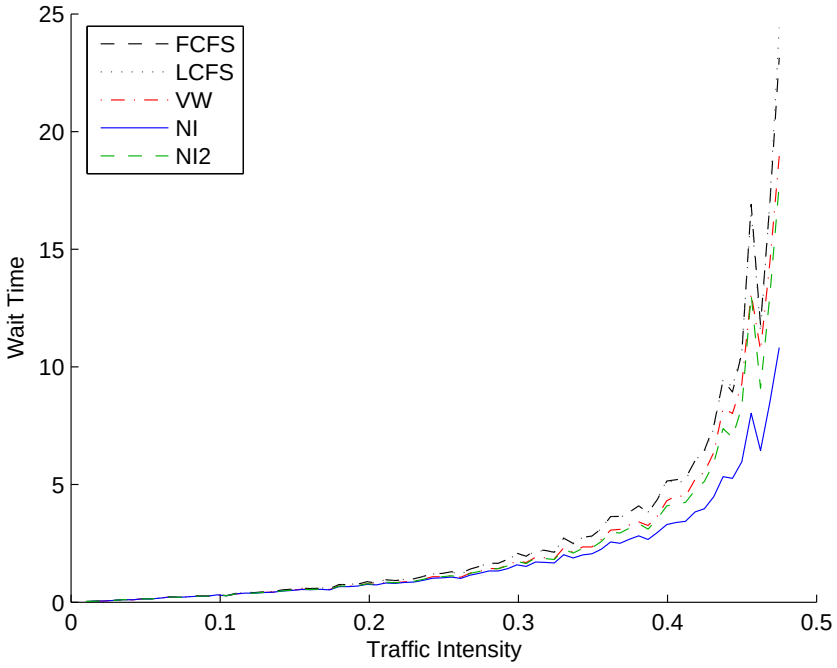


Figure 4.15: Average wait time for three service disciplines with large switching time.

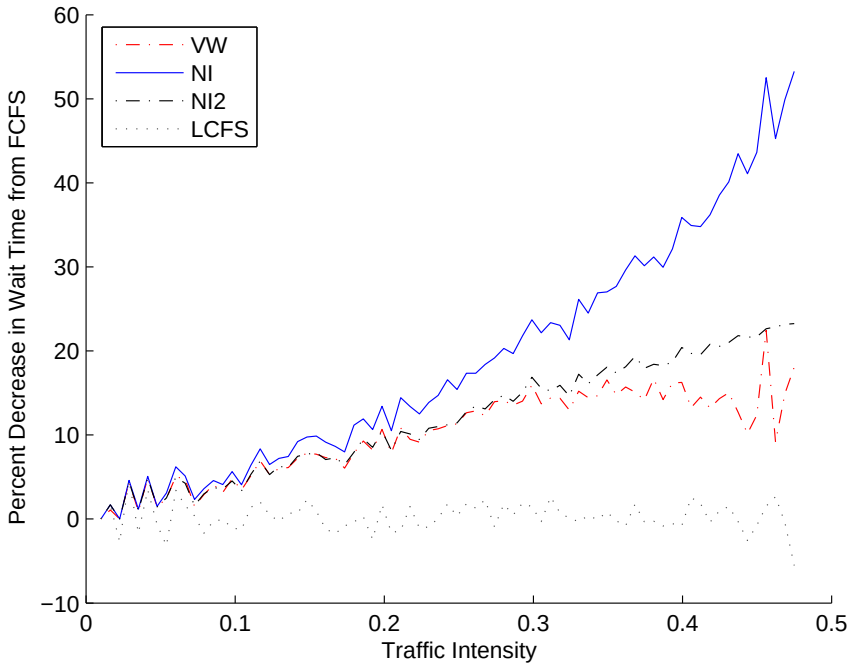


Figure 4.16: The percent decrease in the wait time from the first come first served service rule.

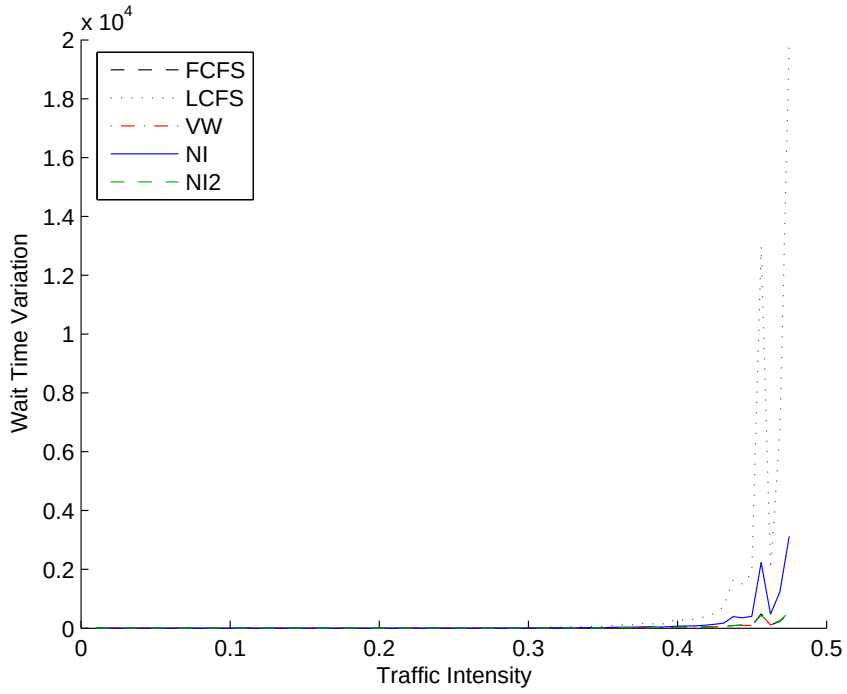


Figure 4.17: Variance of the wait time with large switching time.

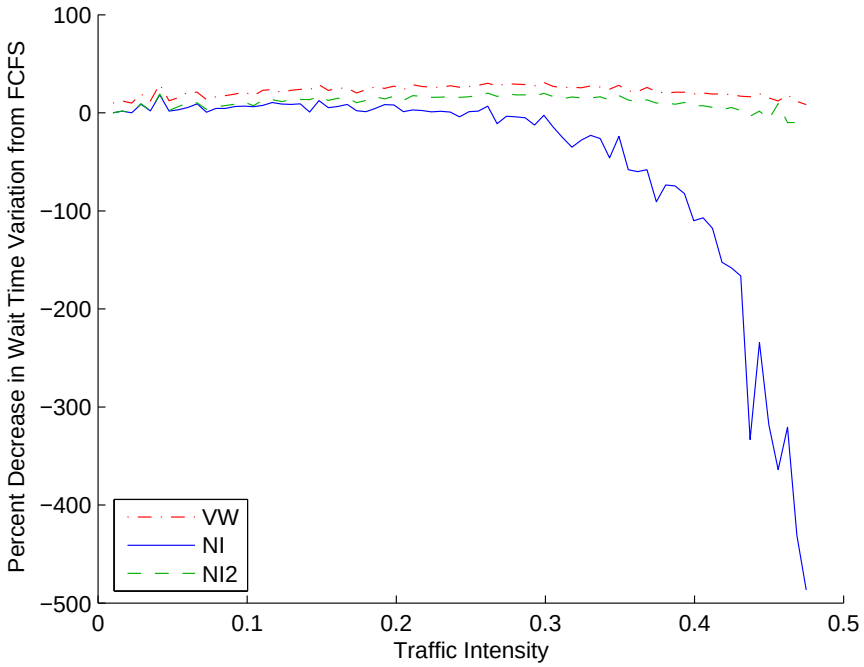


Figure 4.18: The percent decrease in wait time variation from the first come first served service rule.

4.2.3 Large Switching Time

In this section we will look at the system with a switching time that is twice the size of the average service time. As in previous simulations, Figures 4.19 and 4.20 show the the LCFS protocol provides a similar average wait time to the FCFS policies, while the VW, NI2 and NI policies provide the lowest average wait time in that order.

4.2.4 Very Large Switching Time

Figures 4.21 and 4.22 show that the variance of the wait time is decreased by the VW and NI2 disciplines, while the NI discipline makes large sacrifices in variance to gain the low average wait time.

4.2.5 General Comments

As the ratio of the switching time to the average service time increases, it should be noted that while the performance of every discipline becomes worse, the relative performance between the different disciplines stay similar. The notable exception here is the decrease in waiting time given by the NI2, and VW disciplines. Figures 4.8, 4.12, and 4.20 show that these discipline's average wait time in high traffic regions decreases more quickly as the switching time increases than the NI discipline does. Additionally, the VW discipline's wait time, while close to the performance of the NI2 discipline at low traffic intensities, becomes is worse at high traffic intensities, especially as the switching time increases. While both algorithms are minimizing the same quantity, the VW discipline optimizes over a smaller subset of the possible tour permutations. As a consequence, the NI2 algorithm performs better at higher traffic intensities because of the large number of customers in the queue. The difference in wait time at high switching times is due to the greater impact that small changes in service order cause. This stems from the reduced stability range for high switching times.

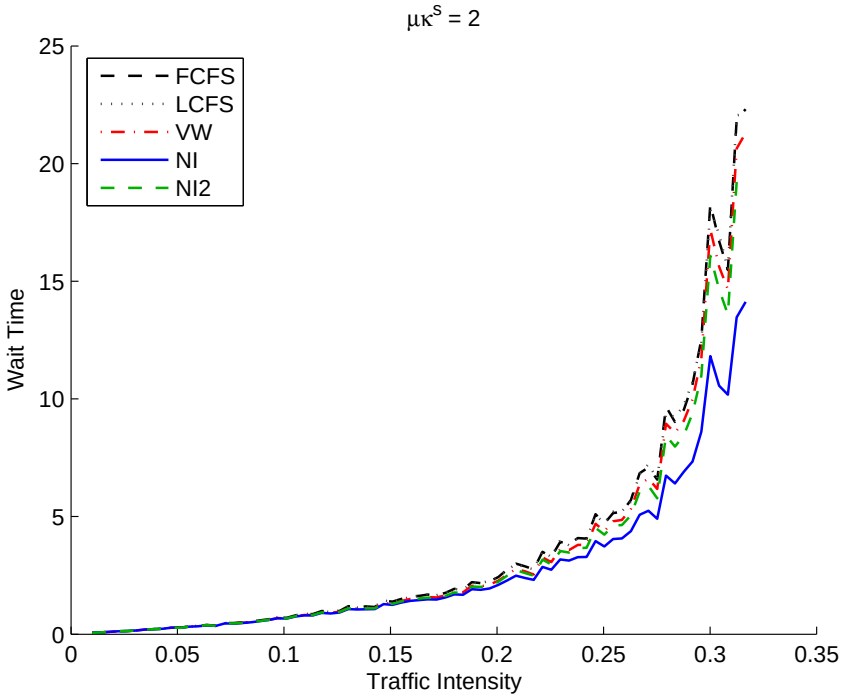


Figure 4.19: Average wait time for three service disciplines with large switching time.

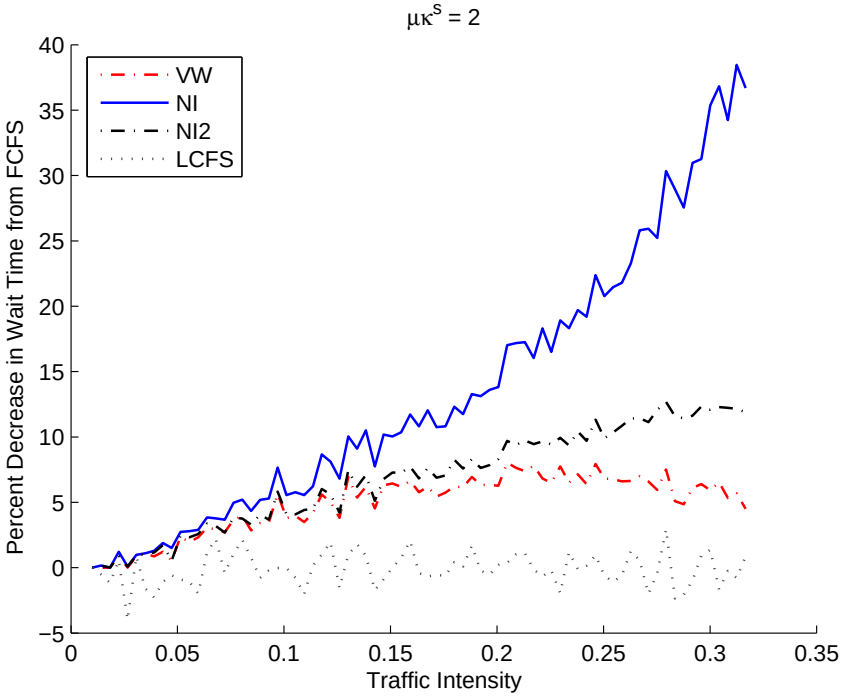


Figure 4.20: The percent decrease in the wait time from the first come first served service rule.

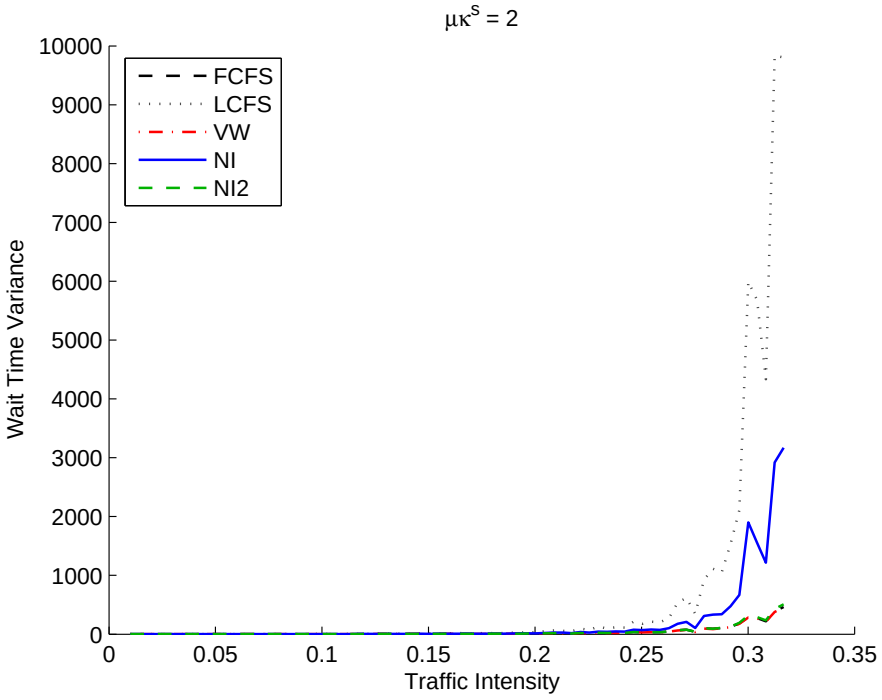


Figure 4.21: Variance of the wait time with large switching time.

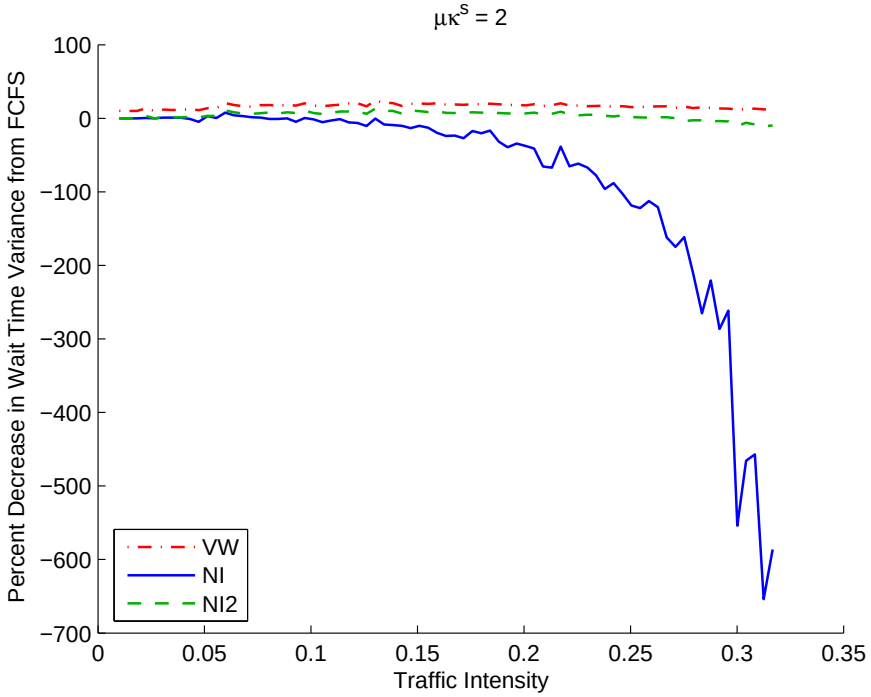


Figure 4.22: The percent decrease in wait time variation from the first come first served service rule.

4.3 Extended Lin-Kernighan

In this section, we will discuss the Extended Lin-Kernighan algorithm in detail. This algorithm is based on the nearest insertion tour creation algorithm, but is applied as a local optimization method. It will be shown in the following that this algorithm is an extension of the Lin-Kernighan algorithm, the most commonly used algorithm for finding low cost tours.

4.3.1 TSP Heuristic Basics

Many combinatorial optimization problems can be expressed as 'find from a set E a subset T that satisfies some criterion C and minimizes an objective function f.' The traveling salesman problem is no exception. Given a graph $G(V, E)$ find a subset of the edges T that form a Hamiltonian tour (condition C) and minimizes the tour length (objective function f). The following is the basis for one approach for heuristically solving combinatorial optimization problems [28]:

i) Generate an initial random or pseudo-random solution T that satisfies condition C.

ii) Attempt to find a transformation T' of the initial solution T that improves the cost

iii) If a solution is found that satisfies $f(T') < f(T)$, where $f(T)$ is the tour cost, then replace T with T' and repeat step *ii)*.

iv) If no improving solution is found, the T is a locally optimal solution.

The objective is to find a transformation rule that minimizes the number of local minima. If a transformation rule yields a unity set of local minima for all initial solutions, that rule is the solution to the combinatorial minimization. There is no known rule for the traveling salesman problem that satisfies this criteria. It is an open question if one exists.

By the nature of the traveling salesman problem, all transformation rules involve the replacement of edges in the current tour with new edges. The only difference between the optimal solution and a suboptimal tour, is that there are up to n segments in the suboptimal tour that are not in the optimal tour. The complication becomes determining the edges that need to be replaced. As with any combinatorics problem, the number of possibilities quickly outstrips our computational ability as the size of the problem increases.

This method for solving combinatorial optimization problems requires two algorithms, a tour creation algorithm, and a local optimization algorithm or transformation. The most commonly used tour construction heuristic is the *nearest neighbor* (NN) heuristic. The algorithm starts at a random city then selects the nearest neighbor not in the tour as the next city. Denoting the length of the optimal tour as L_{opt} and the length of a NN tour L_{NN}, we are guaranteed a bound of

$$\frac{L_{NN}}{L_{opt}} \leq 0.5(\lceil \log_2 N \rceil + 1) \tag{4.1}$$

which grows with the size of the problem.

Figure 4.23 shows how the guarantee on the tour quality decreases as the problem size increases.

The best tour construction heuristic is the *Christofides* heuristic. It guarantees a bound of $L_C/L_{opt} \leq 1.5$ although at a significant increase in running time [29].

One tour creation heuristic that is not used in practice very often is the *nearest insertion* heuristic. This heuristic guarantees a bound of $L_{NI}/L_{opt} \leq 2$, which is not a function of problem size [29]. The NI heuristic starts with three random nodes, then inserts vertices one at a time into the edge of the existing tour that minimizes the overall tour cost.

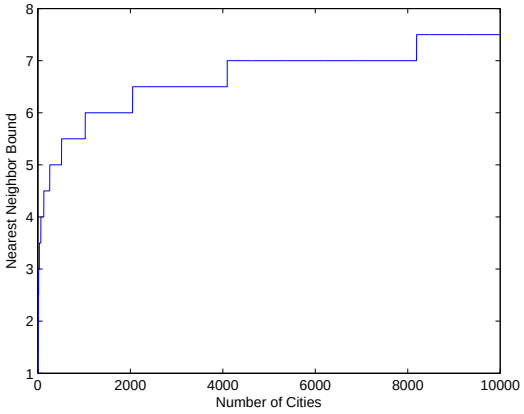


Figure 4.23: Upper bound on the initial nearest neighbor tour quality.

For the laser communications problem, the tour creation algorithm used will be the first- come first-served rule. This rule is used because the most commonly used service discipline is FCFS. Additionally, there is no analog to the nearest neighbor heuristic for the lasercom problem.

4.3.2 Tour Optimality

The foundation for most TSP heuristic is the following necessary condition for the optimality of a tour noted by Merrill Flood [30]. In any optimal tour, $\{t_0, t_1, \ldots, t_{n-1}\}$, for an n-city TSP, for each $0 \leq p < q < n$ we have

$$c_{t_{p-1}t_p} + c_{t_q t_{q+1}} \leq c_{t_{p-1}t_q} + c_{t_p t_{q+1}} \tag{4.2}$$

where the subscripts are taken modulo n. A pair (p, q) that violates (4.2) is an intersecting pair, which means that there are two edges crossing. The objective is to find a series of inversions that create an intersectionless tour. These inversions or 'flips' with an accompanying insertion are known as a *2-opt* move. That is, replacing two edges of tour with two different edges.

4.3.3 Lin-Kernigan

Shen Lin in [31] provide a common framework for describing tours that are optimal with respect to flips and insertion. He defined a tour as *k-optimal* if there does not exist k edges that can be replaced by any other set of k edges and reduce the overall tour cost. A *k-opt* move replaces k edges with a different set of k edges. In [28], Lin and Kernighan suggested a variable *k-opt* algorithm that has become known as the Lin-Kernighan algorithm. This algorithm is a series of flips and insertions that create a $k - opt$ move. The basic algorithm starts with a vertex which will be denoted as the *base*. The the remaining vertices are tested in turn for a flip and insertion. These test vertices are known as the *probe*. If the *base* and *probe* satisfy the inequality

$$c(base, next(base)) + c(probe, next(probe)) \tag{4.3}$$

$$> c(next(base), next(probe)) + c(base, probe) \tag{4.4}$$

then replacing the edges *(base,next(base))* and *(probe,next(probe))* with *(next(base),next(probe))* and *(base,probe)* will improve the total cost of the tour. This is an improving $2 - opt$ move. The Lin-Kernighan algorithm uses a simplified version of this inequality as the test inequality. This inequality is given by:

$$c(base, next(base)) - c(next(base), next(probe)) > 0. \tag{4.5}$$

The algorithm then goes on to make a series of additional flips and insertions to improve the cost of the tour. While some of these intermediate steps may increase the overall cost, in the end, they may lead to an improvement of the tour cost over the initial move*.

4.3.4 Extended Lin-Kernighan

The Extended Lin-Kernighan algorithm is based on the nearest insertion tour construction heuristic. This algorithm iteratively searches each segment of a tour of length k where $1 < k < \lceil n/2 \rceil$ and n is the length of the tour. Given a graph $G = (V, E)$ with a tour T and a segment S given by

*See Appendix II for an extended description of the Lin-Kernighan algorithm.

$(baseA, baseB)$. If an edge *(probe,next(probe))* of the tour $T \setminus S$ satisfies the condition

$$\begin{array}{c} c(baseA, prev(baseA)) + c(baseB, next(baseB)) \\ -c(prev(baseA), next(baseB)) > c(baseA, probe) \\ +c(baseB, next(probe)) - c(probe, next(probe)), \end{array} \tag{4.6}$$

where *baseA* is the first vertex in the segment, *baseB* is the end vertex in the segment, then the segment *(baseA,baseB)* can be inserted at edge *(probe,next(probe))* and decrease the overall cost of the tour.

4.3.5 Relationship to Lin-Kernighan

To compare the Extended Lin-Kernighan algorithm to the Lin-Kernighan algorithm, we will first make some notation changes. The Lin-Kernighan heuristic replaces

$$E(base, next(base)) \rightarrow E(base, probe) \tag{4.7}$$

$$E(probe, next(probe)) \rightarrow E(next(base), next(probe)) \tag{4.8}$$

based on satisfying the inequality

$$\begin{array}{c} c(base, next(base)) + c(probe, next(probe)) > \\ c(base, probe) + c(next(base), next(probe)). \end{array} \tag{4.9}$$

Using the notation of the Lin-Kernighan heuristic, the labels for the nodes are relabeled

$$\begin{aligned} baseA &\rightarrow next(base) \\ baseB &\rightarrow probe \\ prev(baseA) &\rightarrow base \\ next(baseB) &\rightarrow next(probe) \\ probe &\rightarrow test \\ next(probe) &\rightarrow next(test) \end{aligned}$$

where the edge $E(test, next(test))$ is an edge in the tour $\{next(probe), \ldots, base, next(probe)\}$. This condition ensures that the algorithm always ends with a tour. In this label system, the Extended Lin-Kernighan algorithm takes the edges *E(base,next(base))* and *E(probe,next(probe))* and replaces them with edges *E(test,next(base))* and *E(next(test),probe)* base on satisfying the inequality

$$\begin{array}{c} c(base, next(base)) + c(probe, next(probe)) - c(base, next(probe)) > \\ c(next(base), test) + c(probe, next(test)) - c(test, next(test)). \end{array} \tag{4.10}$$

If the vertices *test* and *next(probe)* are the same and vertices *next(test)* and *base* are the same, then the Extended Lin-Kernighan algorithm reduces identically to the *2-opt* move used in Lin-Kernighan.

4.3.6 Reducing Running Time

In the following we will investigate methods for reducing the algorithm's running time. We will reduce the equation size by introducing the shorthand notation $b \equiv base$, $nb \equiv next(base)$, $p \equiv probe$, $np \equiv next(probe)$, $t \equiv test$, and $nt \equiv next(test)$.

Subtracting the two inequalities and requiring that the new algorithm provides shorter tours than the LK algorithm yields the inequality

$$\begin{gathered} c(b,nb)+c(p,np)-c(b,np)+c(t,nt)-c(nb,t)-c(p,nt)\geq \\ c(b,nb)+c(p,np)-c(nb,np)-c(p,b)>0, \end{gathered} \tag{4.11}$$

which reduces to the equation,

$$-c(b,np)+c(t,nt)-c(nb,t)-c(p,nt)\geq -c(nb,np)-c(p,b) \tag{4.12}$$

or equivalently,

$$c(b,np)-c(t,nt)+c(nb,t)+c(p,nt)\leq c(nb,np)+c(p,b). \tag{4.13}$$

This can be further reduced by choosing *base* and *probe* so that the segment to be tested is defined. Equation 4.13 then becomes

$$c(nb,t)+c(p,nt)-c(t,nt)\leq C \tag{4.14}$$

where

$$C=c(nb,np)+c(p,b)-c(b,np) \tag{4.15}$$

is a constant.

The Lin-Kernighan algorithm only has to find a set of *probe* vertices that will improve the tour length. The condition given is:

$$c(b,nb)-c(nb,np)>0 \tag{4.16}$$

which is the cost to improve one of the edges. For the new algorithm, two sets of promising vertices for *probe* and *test* are needed.

By (4.15), we see that a good edge is an edge that is long, with its endpoints close to to the ends of the segment. So instead of choosing a segment, it would be beneficial to choose long edge that is not connected to its closest neighbors and find the segment closest to its two endpoints. By using the (4.15) we can reduce the overall number of edges that need to be checked, reducing the overall computation time.

Chapter 5

Low-Earth Orbit Satellite Constellations

In the following we will present work on autonomous satellite scheduling for constellations of low-earth orbit satellites. This work was the beginning of the study of the laser communications system. The problem investigated here, is how a group of satellites can determine optimal orders of service for the entire group when there are multiple satellites that can serve the same terminals. Aspects of this work will be applied to the laser communications problem when multiple lasercom satellites are being used to fulfill the same mission with shared terminals. In the following we will show how continuous global coverage can be simulated.

5.0.7 Low-Earth Orbit Satellite Constellations

For many applications, such as RF communications, continuous global coverage is needed. Currently the only realistic method for achieving continuous global coverage is through a constellation of satellites with wide beamwidths. Systems such as the GlobalStar satellite telephone network and the Global Positioning System are practical examples of this. The wide beamwidth allows the constellation to be built with a reasonable number of satellites. The GlobalStar network consists of 52 satellites orbiting in low-earth orbit at an altitude of approximately 1,400 km. The GPS network consists of 29 satellites orbiting in medium-earth orbit at approximately 20,000 km.

However, for applications with narrow beamwidths such as imaging and laser communications, the number of satellites needed for continuous global coverage is prohibitive. A constellation of satellites at an altitude of 1,400 km with a beamwidth of 15 degrees would require at least 1,150 satellites to achieve continuous global coverage.

In reality, continuous global coverage is not necessary in most situations. At any instant in time, only a fraction of the surface of the earth needs coverage. It is sufficient to have the ability to point a satellite at any location on the earth at any time with enough satellites to handle the user load. This allows the simulation of continuous global coverage with narrow beamwidth satellites.

Mathematical Preliminaries

Let $\mathbb{R}$ (resp., $\mathbb{Z}$) denote the set of real numbers (resp., positive integers), let $\mathbb{R}^{n\times m}$ (resp., $\mathbb{Z}^{n\times m}$ denote the set of real (resp., positive integer) matrices, and let A^{T} denote the transpose of A. We write e or e_W for denoting $e = [1, 1, 1, \ldots]^{\mathrm{T}}$. Next we use the notation $x \geq\geq y$ (resp. $x >> y$) to denote that $x_i \geq y_i$ (resp. $x_i > y_i$), $i = 1, \ldots, n$, where $x, y \in \mathbb{R}^n$. Finally, for $A, B \in \mathbb{R}^{m\times n}$, we write $A \circ B$ to denote the Hadamard product of A and B.

Let $G = (V, E)$ denote a *bipartite graph* where $V = X \times Y$ is the set of vertices and E is the set of edges. It is convenient to assume that the elements of X and Y are enumerated for $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$ so that X and Y may be assumed to belong to $\mathbb{Z}^n$ and $\mathbb{Z}^m$, respectively. Furthermore, E can be represented by the connectivity (or adjacency) matrix given by

$$E_{ij} = \begin{cases} 1, & \text{if } (i,j) \text{ is an edge} \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

for all $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$. Next, let $G = (V, E, W)$ denote a *weighted bipartite graph* where (V, E) is a bipartite graph with $W \in \mathbb{R}^{n \times n}$ denoting the weight matrix associated with the edges. We assume that $W_{ij} \geq 0$, if $E_{ij} = 1$ and $W_{ij} = \infty$ if $E_{ij} = 0$.

A vertex $i \in X$ (resp., $j \in Y$) is *incident* to an edge if there exists $j \in Y$ (resp., $i \in X$) such that (i, j) is an edge. In this case, (i, j) are *adjacent*. A set of vertices $U \subseteq V$ is a *vertex cover* of G if the set $\{(i, j) : E_{ij} = 1, i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}\} \subseteq U$ that is, if every edge in E is incident with a vertex in U. A *minimum vertex cover* is a vertex cover of G with the minimum number of vertices. The existence of such a minimum vertex cover is easy to establish.

Two edges of a graph are *independent* if they do not have any common incident vertices. A *matching* of G is a set $M \subseteq E$ which is a *subgraph* of independent edges in a graph $G = (V, E)$. Since G is bipartite it follows that the set of incident vertices U of a matching will be of the form $\hat{X} \times \hat{Y}$ such that $\hat{X}$ and $\hat{Y}$ have the same number of vertices. A *maximum matching* of G is a matching of G that has the largest number of edges (and hence the largest number of incident vertices). A *perfect matching* of G is a (maximum) matching of G such that the set of incident vertices is $X \times Y$ (see Figure 5.1).

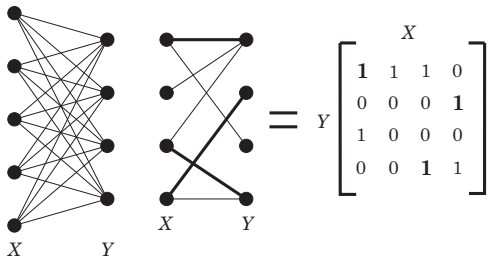


Figure 5.1: A complete bipartite graph, a matching on a bipartite graph and the equivalent edge matrix

By definition, if a graph G has a perfect matching M then $m = n$, that is X and Y have the same number of vertices. Furthermore, $Me = M^{\mathrm{T}}e = e$. Hence, it follows that if a bipartite graph $G = (V, E)$ has a perfect matching then $Ee \geq\geq e$ and $E^{\mathrm{T}}e \geq\geq e$. However, note that these conditions are not sufficient (for example, consider the graph given in Figure 5.2). The following result provides a necessary and sufficient condition for the existence of a perfect matching. First, however, we introduce the following notation. Let $\mathcal{N}_D : \mathbb{R}^{m \times n} \to \mathbb{R}$ be defined by

$$\mathcal{N}_D(E) \triangleq e_m^{\mathrm{T}} D e_n \tag{5.2}$$

where $D \in \mathbb{R}^{m \times n}$ is given by

$$D_{ij} = \begin{cases} E_{ii}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{5.3}$$

Note that if $m = n$ then $\mathcal{N}_D(E)$ denotes the number of ones on the diagonal of E. Next, let $\nu_{\max} : \mathbb{R}^{m\times n} \to \mathbb{R}$ be defined by

$$\nu_{\max}(E) \triangleq \max_{P_m\in\Pi_m, P_n\in\Pi_n} \mathcal{N}_D(P_m E P_n) \tag{5.4}$$

where $\Pi_m \subseteq \mathbb{R}^{m\times m}$ and $\Pi_n \subseteq \mathbb{R}^{n\times n}$ denote the set of $m \times m$ and $n \times n$ *permutation matrices*, respectively. The quantity $\nu_{\max}$ is equivalent to the size of the minimum vertex cover.

Theorem 5.0.1 *Consider a bipartite graph $G = (V, E)$ where $E \in \mathbb{R}^{m\times n}$. The size of the maximum matching is $\nu_{\max}(E)$. Furthermore, if $m = n$ then G has a perfect matching if and only if $\nu_{\max} = n$.*

Proof. Let M denote a maximum matching of G. In this case it is easy to show that there exist permutations of X and Y such that the edge matrix M_Π of the permutations X_Π and Y_Π is such that $M_{\Pi_{ij}} = 0$, $i \neq j$ and the size of the matching M_Π is given by $e_m^{\mathrm{T}} M_\Pi e_n = e_m^{\mathrm{T}} M e_n$. The result now follows from the fact that there exist $P_m \in \Pi_m$ and $P_n \in \Pi_n$ such that $M_\Pi = P_m M P_n$ and $\mathcal{N}_D(P_m E P_n) = e_m^{\mathrm{T}} M_\Pi e_n$.

Next, if $m = n$ it follows that G has a perfect matching if and only if $\nu_{max}(E) = n$ by noting that a perfect matching is a maximum matching of size n. □

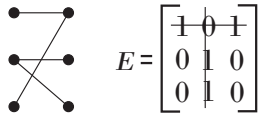


Figure 5.2: A minimum line covering on an edge matrix

Remark 5.0.1 *It follows from Königs theorem [32] that $\nu_{\max}(E)$ is the size of the minimum vertex cover. Hence, $\nu_{\max}(E)$ denotes the minimum number of lines (drawn across rows and columns) that are sufficient to cover all the 1's in E which implies that $\nu_{\max}(E)$ is the maximum number of 1's, no two of which are in the same line (row or column) of E (see Figures 5.2 and 5.3).*

Remark 5.0.2 *Although Theorem 5.0.1 gives necessary and sufficient conditions for the existence of a perfect matching it is in general computationally more efficient to use Königs theorem to compute $\nu_{\max}(E)$.*

Next, we consider the *virtual perfect matching problem* where given a graph $G = (V, E)$ that does not have a perfect matching, we provide a method for constructing the *virtual graph* $\hat{G} = (\hat{V}, \hat{E})$ such that $\hat{G}$ has a perfect matching and $\hat{G}$ is a *projection* of G, that is, G is a subgraph of $\hat{G}$ such that when all the vertices in $\hat{V}\backslash V$ are eliminated (along with their edges) then the resultant graph is G. Note that if $E \in \mathbb{R}^{m\times n}$ then by choosing $\hat{V} = \hat{X} \times \hat{Y} \subseteq \mathbb{R}^{m+n} \times \mathbb{R}^{m+n}$ such that all the m additional (virtual) vertices of $\hat{X}$ and the n additional virtual vertices of $\hat{Y}$ are connected with every other vertex of $\hat{Y}$ and $\hat{X}$, respectively, it can be shown that $\hat{G}$ has a perfect matching. While this is true for all graphs, the size of the graph doubles, and often there exists a perfect matching with a smaller number of virtual vertices added. The following result provides a construction $\hat{G}$ that has the minimum number of virtual vertices.

Theorem 5.0.2 *Let $G = (V, E)$ be a bipartite graph. Then there exists $\hat{G} = (\hat{V}, \hat{E})$ such that $\hat{G}$ is a projection of G, $\hat{G}$ has a perfect matching, and*

$$\nu_{\max}(\hat{E}) = (m + n) - \nu_{\max}(E) \tag{5.5}$$

Furthermore, there does not exist $\tilde{G} = (\tilde{V}, \tilde{E})$ such that $\tilde{G}$ is a projection of G, $\tilde{G}$ has a perfect matching, and $\nu_{\max}(\tilde{E}) < \nu_{\max}(\hat{E})$.

Proof. Let M be a maximum matching and assume without loss of generality that $M_{ij} = 0$, $i \neq j$. Now, it can be shown that by adding $n - \nu_{\max}(E)$ and $m - \nu_{\max}(E)$ virtual vertices to X and Y the resultant $\hat{G}$ has a perfect matching. The minimality of $\nu_{\max}(\hat{E})$ given by (5.5) can be shown in a similar manner. □

Remark 5.0.3 *Note that if G has a perfect matching, that is, $\nu_{\max}(E) = m = n$ is follows that $\nu_{\max}(\hat{E}) = \nu_{\max}(E)$ and hence $G = \hat{G}$ (since $\hat{G}$ is a projection of G)*

Remark 5.0.4 *As illustrated in Figure 5.3, the existence of a perfect matching can be ensured by adding virtual vertices to X and Y where black is the original graph and grey is the added vertices and edges. In this case $\nu_{\max}(E) = 2$ while $m = n = 3$, hence the original graph does not have a perfect matching. By adding a virtual vertex to X and Y, $\nu_{\max}(\hat{E}) = 4$ and $\hat{m} = \hat{n} = 4$ hence $\hat{E}$ has a perfect matching. Figure 5.3 shows how the edge matrix can be augmented to provide a perfect matching.*

Finally, we consider the *optimal assignment problem*. Specifically, let $G = (V, E, W)$ be a weighted bipartite graph that has at least on perfect matching. The optimal assignment problem is the problem of finding a perfect matching that has the *minimum weight*, that is, finding M such that

$$\overline{w} = \min_{M \in \mathcal{M}} e^{\mathrm{T}}(W \circ M)e \tag{5.6}$$

where $\mathcal{M}$ denotes the set of all perfect matchings of G.

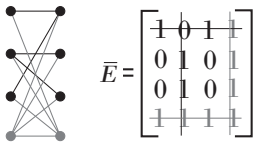


Figure 5.3: Augmented edge matrix with the new minimum line cover

More generally, let $G = (V, E, W)$ be a weighted bipartite graph such that $E \in \mathbb{R}^{m \times n}$ which may not have a perfect matching. In this case, the optimal assignment problem is given by

$$\overline{w} = \min_{\hat{M} \in \hat{\mathcal{M}}} e^{\mathrm{T}}(\hat{W} \circ \hat{M})e \tag{5.7}$$

where $\hat{M}$ denotes the set of perfect matchings of $\hat{G}$, the virtual graph of G, and $\hat{W}$ denotes the weighting matrix of $\hat{G}$ such that $\hat{W}_{ij} = 0$ if i or j correspondes to a virtual vertex and $\hat{W}_{ij} = W_{ij}$ otherwise.

Note that the construction of $\hat{G}$ can be performed using the algorithm described in [33] and the optimal assignment problem given by (5.7) can be solved by using the Hungarian method [32–34].

Optimal Dynamic Assignment Problem

In this section we will define the problem of minimizing the total edge weight for a dynamic graph. We will also discuss the conditions for applying this algorithm to a dynamic graph.

A *dynamic graph* is a graph where the vertices, vertex weights, edges, and edge weights are changing with time [35]. Since we are dealing with a matching on a bipartite graph, vertex weights are not used and can be assumed constant. We will denote a dynamic graph by

$\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t), \mathcal{W}(t))$ where $\mathcal{E}(t)$ and $\mathcal{W}(t)$ are $n \times n$ matrices. As shown in the previous section, if the graph is not square, it can be virtually projected to a square graph on which a perfect matching exists.

In the static case, the edge weight matrix is a matrix of positive constants. In the dynamic case $\mathcal{W}$ depends on time and the time-varying edge matrix. This leads to the general definition of a time-varying weight matrix for a bipartite graph given by

$$\mathcal{W}_{ij}(t, \mathcal{E}(t)) \triangleq \begin{cases} f_{ij}(t), & \mathcal{E}_{ij}(t) = 1 \\ \infty, & \mathcal{E}_{ij}(t) = 0 \end{cases} \tag{5.8}$$

where $f_{ij}(t) : \mathbb{R} \to \mathcal{I} \subset \mathbb{R}_+$ are a set of functions that describe the weight for a particular edge of $\mathcal{G}$ and are not necessarily smooth or continuous, and $\mathcal{I}$ is a closed bounded interval.

The choice of the weight function $f_{ij}(t)$ is determined by the problem objective. In the case of LEO sensing or imaging constellations, one objective is to maximize the total time that the satellites are tracking targets. This is equivalent to minimizing the total time that the satellites spend moving between targets. For a communications satellite, one objective would be to maximize the bandwidth, another would be to prevent buffers from overflowing, resulting in data loss.

In the following we will show that the discrete time algorithm approaches the continuous time solution for a LEO imaging constellations. With a weight function that is proportional a satellite's slew time for a given target, the continuous optimal assignment problem is given by

$$\hat{w} = \int_{t_1}^{t_s} \min_{M \in \mathcal{M}} w(M) \mathrm{dt} \tag{5.9}$$

$$= \int_{t_1}^{t_s} \min_{M \in \mathcal{M}} e^T (\mathcal{W}(t) \circ M) e \tag{5.10}$$

where $\mathcal{M}(\mathcal{E})$ is the set of admissible matchings at time t.

Equations (5.9),(5.10) shows that the behavior of the total weight function is dependent on the matching and the edge weight functions. For example, if the matching does not change on the time interval and the edge weight functions are continuously differentiable, then the total weight function will be continuously differentiable on the interval. On the other hand, if the edge weight function is constant on the time interval, the matching will only change when the edge matrix changes. This means that the matching and edge weight functions must be "well behaved" for the integral to behave nicely.

In the following we show that if the total edge weight is bounded on the time interval, then the solution is optimal and the finite time version of the problem approaches the optimal solution in the limit. This is equivalent to a perfect matching existing and $f_{ij}(t)$, $i, j = 1, 2, \ldots, n$ bounded on $[t_1, t_s]$.

In actual implementation, the algorithm will be applied in a discrete manner with the matching changing only at discrete times. In the following we show that as the time between the matching updates decreases, the algorithms behavior approaches the continuous time solution.

We will start by defining a discrete time version of the optimal assignment problem. We will allow the matching to be made at discrete times in the time interval. Let $P = \{t_k\}_{k=1}^{s}$ be a sequence of times that partition the interval (t_1, t_s). Now let

$$w^k(M) \triangleq e^{\mathrm{T}} (\mathcal{W}(t_k) \circ M) e \tag{5.11}$$

be the total edge weight of a matching at time t_k where M is an admissible matching. Let the total edge weight $w^k(M)$ remain constant on the interval $[t_k, t_{k+1})$. The minimum weight matching $\overline{M}(t_k)$ is calculated at the beginning of each time interval and remains constant over the interval and is given by

$$\overline{w}(\overline{M}(t_k)) \triangleq \min_{M \in \mathcal{M}} w^k(M) = \min_{M \in \mathcal{M}} e^{\mathrm{T}}(\mathcal{W}(t_k) \circ M)e \tag{5.12}$$

The total weight of the matching over the interval is the weight at the beginning of each subinterval times the length of the interval, and the optimization problem becomes

$$\begin{aligned} \hat{w} &= \sum_{k=0}^{s-1} \min_{M \in \mathcal{M}} w^k(M)(t_{k+1} - t_k) && (5.13) \\ &= \sum_{k=0}^{s-1} \min_{M \in \mathcal{M}} e^{\mathrm{T}}(\mathcal{W}(t_k) \circ M)e(t_{k+1} - t_k) && (5.14) \\ &= \sum_{k=0}^{s-1} \overline{w}(t_k)(t_{k+1} - t_k) && (5.15) \end{aligned}$$

Next, to show that as the partition gets finer, the discrete-time problem approaches the continuous time problem, we must show that the total edge weight is Riemann integrable. Let

$$I(g) \triangleq \lim_{\|P\| \to 0} \sum_{k=0}^{s-1} \overline{w}(t_k)(t_{k+1} - t_k) \tag{5.16}$$

where $\|P\| = \max_k |t_k - t_{k-1}|$. Since, by definition, $\overline{w}(t_k)$ is constant and finite on the interval $[t_k, t_{k+1})$, $I(g)$ exists and is finite. Hence, by Riemann integrability

$$\begin{aligned} \hat{w} &= \int_{t_1}^{t_s} \min_{M \in \mathcal{M}} w(M)\mathrm{dt} && (5.17) \\ &= \lim_{\|P\| \to 0} \sum_{k=1}^{s} \overline{w}(t_k)(t_k - t_{k-1}) && (5.18) \end{aligned}$$

which shows that in the limit, the discrete time case approaches the continuous time optimal solution. Figure 5.4 show a simulation of a typical satellite constellation.

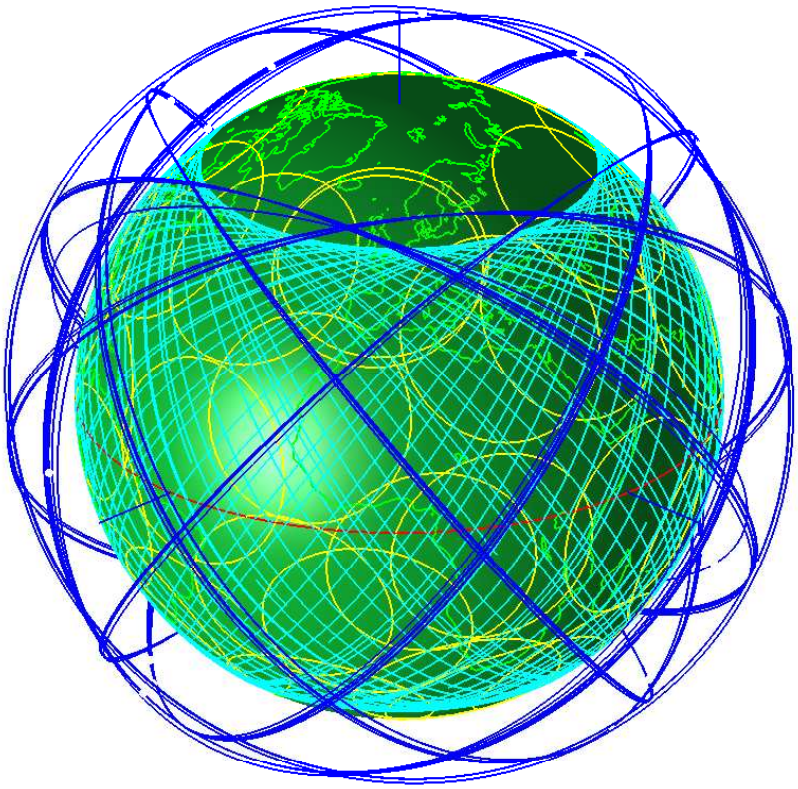


Figure 5.4: GlobalStar Satellite Constellation

Chapter 6

Conclusion

6.1 Conclusion

Laser communications satellites will one day be an important part of the global communication infrastructure. Because of their unique physics, they provide challenges that do not exist in the traditional radio frequency satellites. Laser communications satellites are most efficiently used as a dedicated connection. Used in this manner, the satellites can be operated closer to their maximum bandwidths without the system becoming unstable. However, some applications and classes of users require connections at their convenience without concern being concerned about their position. Applications such as transferring images from a GlobalHawk UAV to a command center require that the laser communications satellite adapt. While this class of users should be kept to a minimum, applying the optimization methods developed in this dissertation will allow the satellite to provide reasonable bandwidth and quality of service, without reducing the satellites overall capability unduely.

6.2 Contributions

The main contributions of this dissertation are a mathematical definition of the bandwidth-on-demand laser communications problem, both from an system perspective and a solution perspective. Additionally, we have defined three new cost functions that show marked improvement in reducing the system time and wait time over the traditional service disciplines such as first-come first-served. Next, we have analyzed the system from a queueing theory perspective and derived equations that describe the average behavior of the system wait time and system time along with the variance of those parameters. We also show that these equations more accurately describe the average system behavior than the best previously known bounds derived by Bertsimas and Van Ryzin. Next, we show that the cost functions presented in this dissertation provide better performance than the standard service disciplines. Next, we present a new algorithm for solving the traveling salesman problem that is easily applied to the systems such a bandwidth-on-demand lasercom, whose behavior cost cannot be described by a cost matrix that characterizes a standard traveling salesman problem. Finally, we present a system for emulating global coverage with a constellation of narrow beamwidth satellites.

6.3 Future Work

There are many directions and branches to be considered in the solution to this problem. One immediate area of interest is to find bounds on the waiting time and waiting time variance in a queueing system with delay for service disciplines different than FCFS. Another problem of interest would be to look at the system with state dependent switching times from an analytical standpoint. While the nearest insertion algorithm can be applied to give good performance, there are no analytical guarantees. Another expansion of this research is to apply this system to a group of laser communications satellites or satellites with multiple laser terminals. This would involve finding a cooperative service discipline that is robust to failure are requires very little information exchange between satellites. A further area of interest is the reaction of the system to varying rate parameters.

Finally, another interesting line of research would be the application of compartmental system theory to the dynamic equation of motion for the queue size given by equations 3.16 and 3.16. This system can be thought of as a compartmental system with statistical rates of transfer between compartments. This could shed new light on the dynamic behavior of queueing systems. Currently, only a queueing system with Poisson arrival and service processes and a first come first served service discipline can be analyzed dynamically and even then with great difficulty.

Bibliography

Bibliography

[1] K. E. Wilson and J. R. Lesh, "Overview of the Galileo Optical Experiment (GOPEX)," *TDA Progress Report 42-114*, pp. 192–204, 1992.

[2] K. E. Wilson, J. R. Lesh, A. Kenichi, and A. Yoshinori, "Preliminary results of the Ground/Orbiter Lasercomm Demonstration experiment between table mountain and the ets-vi satellite," *SPIE Free-Space Laser Communication Technologies VIII*, vol. 2699, pp. 121–132, 1996.

[3] Z. Sodnik, B. Furch, and H. Lutz, "Free-space laser communication activities in europe: SILEX and beyond," *IEEE Lasers and Electro-Optics Society*, pp. 78–79, 2006.

[4] Boeing, *(August 1, 2006)*.

[5] M. Maher, H. Garcia, A. Das, and J. Meadows, "Operations timeline for TSAT planning," in *Military Communications Conference*, 2006, pp. 1–7.

[6] F. Yegenoglu, D. Voce, and D. Gokhale, "TSAT advanced network services and routing architecture," in *IEEE Aerospace Conference*, 2006, pp. 1–6.

[7] T. Le-Ngoc and J. I. Mohammed, "Combined free/demand assignment multiple access (CFDAMA) protocols for packet satellite communications," in *IEEE Conference on Universal Personal Communications*, Ottowa Canada, 1993, pp. 824–828.

[8] E. H. Kjeldsen, "Optimizing satellite communications access with simulation," in *Winter Simulation Conference*, 1994, pp. 1443–1446.

[9] T. Le-Ngoc and S. V. Krishnamurthy, "Performance of combined free/demand assignment multiple access (CFDAMA) protocols with pre-assigned request slots in integrated voice/data satellite communications," in *IEEE*, 1995, pp. 1572–1576.

[10] S. Arnon, S. R. Rotman, and N. S. Kopeika, "Bandwidth maximization for satellite laser communication," in *IEEE Transactions on Aerospace and Electronic Systems*, 1999.

[11] S. Arnon and A. Sason, "Enhanced atm for satellite laser communication networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 3, pp. 1071–1076, July 1999.

[12] M. Allman, D. Glover, and L. Sanchez, "Enhancing TCP over satellite channels using standard mechanisms," *The Internet Society*, January 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2488.txt

[13] A. M. Richard, B. A. Kachmar, T. Fabian, and R. J. Kerczewski, "Overview of In-Space Internet Node Testbed (ISINT)," *NASA*, August 2000. [Online]. Available: http://gltrs.grc.nasa.gov/GLTRS

[14] L. P. Clare, J. R. Agre, and T. Yan, "Considerations on communications network protocols in deep space," in *IEEE*, 2001, pp. 2–943 to 2–950.

[15] G. Açar and C. Rosenbery, "Weighted fair bandwidth-on-demand (WFBoD) for geostationary satellite systems," *SPIE Modeling and Design of Wireless Networks - Invited Paper*, vol. 4531, pp. 218–227, 2001.

[16] F. D. Priscoli and A. Pietrabissa, "Resource management for ATM-based geostationary satellite networks with on-board processing," *Computer Networks*, vol. 39, pp. 43–60, 2002.

[17] Özgür B Akan, J. Fang, and I. F. Akyildiz, "Performance of TCP protocols in deep space communication networks," *IEEE Communications Letters*, vol. 6, no. 11, pp. 478–480, November 2002.

[18] T. Le-Ngoc, V. Leung, P. Takats, and P. Garland, "Interactive multimedia satellite access communications," *IEEE Communications Magazine*, pp. 78–85, July 2003.

[19] F. D. Priscoli and A. Pietrabissa, "Design of a bandwidth-on-demand (BoD) protocol for satellite networks modelled as time-delay systems," *Automatica*, vol. 40, pp. 729–741, 2004.

[20] P. D. Mitchell, D. Grace, and T. C. Tozer, "Burst targeted demand assignment multiple-access for broadband internet service delivery over geostationary satellite," *IEEE Transactions on Selected Areas in Communications*, vol. 22, no. 3, pp. 576–558, April 2004.

[21] H. N. Psaraftis, *Dynamic Vehicle Routing Problems*. New York: Elsevier Science Publishers B.V., 1988.

[22] ——, "Dynamic vehicle routing: Status and prospects," *Annals of Operations Research*, vol. 61, pp. 143–164, 1995.

[23] D. J. Bertsimas and G. van Ryzin, "A stochastic and dynamic vehicle routing problem in the Euclidean plane," *Operations Research*, vol. 39, no. 4, pp. 601–615, 1991.

[24] ——, "Stochastic and dynamic vehicle routing problem in the Euclidean plane with multiple capacitated vehicles," *Operations Research*, vol. 25, pp. 60–76, 1993.

[25] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1984.

[26] L. Breuer and D. Baum, *An Introduction to Queueing Theory and Matrix-Analytic Methods*. Dordrecht: Springer, 2005.

[27] L. Kleinrock, *Queueing Systems. Vol. I: Theory*. New York: John Wiley & Sons, 1975.

[28] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations Research*, vol. 21, pp. 498–516, 1973.

[29] D. S. Johnson and L. A. McGeoch, "The travelling salesman problem: A case study in local optimization," in *Local Search in Combinatorial Optimization*, E. H. L. Aarts and J. K. Lenstra, Eds. Princton University Press, 2003.

[30] M. M. Flood, "The traveling-salesman problem," *Operations Research*, vol. 4, pp. 61–75, 1956.

[31] S. Lin, "Computer solutions of the traveling salesman problem," *The Bell System Technical Journal*, vol. 44, pp. 2245–2269, 1965.

[32] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms.* New York: Wiley, 1992.

[33] G. V. Shenoy, *Linear Programming: Methods and Applications.* New York: Wiley, 1989.

[34] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist. Quarterly*, vol. 2, pp. 83–97, 1955.

[35] F. Harary and G. Gupta, "Dynamic graph models," *Mathematical Computer Modelling*, vol. 25, no. 7, pp. 79–87, 1997.

[36] C. M. Grinstead and J. L. Snell, *Introduction to Probability.* Providence: American Mathematical Society, 1997.

[37] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, *The Traveling Salesman Problem: A Computational Study.* New Jersey: Princton University Press, 2006.

[38] P. C. Mahalanobis, "A sample survey of the acreage under jute in bengal," *Sankhya, The Indian Journal of Statistics*, vol. 4, pp. 511–530, 1940.

[39] J. Beardwood, J. H. Halton, and J. M. Hammersley, "The shortest path through many points," *Proceedings of the Cambridge Philisophical Society*, vol. 55, pp. 299–327, 1959.

[40] D. S. Johnson, L. A. McGeoch, and E. E. Rothberg, "Asymptotic experimental analysis for the Held-Karp traveling salesman bound," in *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, New York, 1996, pp. 341–350.

[41] C. L. Valenzuela and A. J. Jones, "Estimating the Held-Karp lower bound for the geometric TSP," *European Journal of Operational Research*, vol. 102, pp. 157–175, 1997.

[42] M. X. Goemans and D. J. Bertsimas, "Probabilistic analysis of the Held and Karp lower bound for the Euclidean traveling salesman problem," *Mathematics of Operations Research*, vol. 16, pp. 72–89, 1991.

[43] A. P. Punnen, "The traveling salesman problem: Applications, formulations, and variations," in *The Traveling Salesman Problem and Its Variations.* Dordrecht, The Netherlands: Kluwer Academic Publishers, 2002.

[44] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, pp. 48–50, 1956.

[45] R. C. Prim, "Shortest connection networks and some generalizations." *Bell Systems Technical Journal*, vol. 36, pp. 1389–1401, 1957.

[46] N. Christofides, "Bounds for the traveling salesman problem," *Operations Research*, vol. 20, no. 5, pp. 1044–1056, 1972.

[47] M. Bellmore and J. C. Malone, "Pathology of traveling-salesman subtour elimination algorithms," *Operations Research*, vol. 19, no. 2, pp. 278–307, 1971.

[48] N. Christofides, *Graph Theory An Algorithmic Approach.* New York: Academic Press, 1975.

[49] R. Bellman, *Dynamic Programming.* Princeton, New Jersey: Princeton University Press, 1957.

[50] M. Held and R. M. Karp, "A dynamic programming approach to sequencing problems." *Journal of the Society of Industrial and Applied Mathematics*, vol. 10, pp. 196–210, 1962.

[51] M. Padberg and G. Rinaldi, "A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM Review*, vol. 33, pp. 60–100, 1991.

[52] G. Reinelt. [Online]. Available: http://www.iwr.uni-heidelberg.de/groups/comopt/soft/TSPLIB95/TSPLIB.html

[53] G. A. Croes, "A method for solving traveling-salesman problems." *Operations Research*, vol. 6, pp. 791–812, 1958.

Appendix

Appendix I: Statistical Definitions

In this appendix we will review some selected basic statistical information which forms the foundation for some of the discussions of the laser communications system.

Basic Definitions

For a random variable, a description of the possible outcomes over the entire domain of the variable is described by the cumulative distribution function.

Definition 6.3.1 (PDF) *A **probability distribution function** for a random variable X is given by*

$$F_X(x) \triangleq P[X \leq x] \tag{6.1}$$

where $P[X \leq x]$ is the probability that $X \leq x$ for all values of $x \in (-\infty, \infty)$.

For the purposes of calculation, a more useful function is the derivative of the CDF or the probability density function.

Definition 6.3.2 (pdf) *The **probability density function** for a random variable X is given by:*

$$f_X(x) \triangleq \frac{\mathrm{d}F_X(x)}{\mathrm{d}x} \tag{6.2}$$

The laser communications system involves multiple random variables such as the number of users in the system or the amount of time that a user needs the connection. This requires the use of joint probabilities. A *joint* probability distribution function is given by:

$$F_{XY}(x, y) \triangleq P[X \leq x, Y \leq y] \tag{6.3}$$

Likewise, the *joint* probability density function is given by the relation

$$f_{XY}(x, y) \triangleq \frac{\mathrm{d}^2 F_{XY}(x, y)}{\mathrm{d}x\mathrm{d}y} \tag{6.4}$$

For the laser communications system, we will consider the arrival of demands for service and the time it takes to service those demands independent.

Definition 6.3.3 *Two random variables X and Y are **independent** if*

$$F_{XY}(x, y) = F_X(x)F_Y(y) \tag{6.5}$$

which implies that

$$f_{XY}(x, y) = f_X(x)f_Y(y) \tag{6.6}$$

The *expected value* of a continuous random variable is given by:

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) \mathrm{d}x \tag{6.7}$$

where it is assumed that $f_X(x)$ exists and is integrable. For a discrete random variable, the expected value is simply,

$$E[X] \triangleq \sum_{-\infty}^{\infty} x f_X(x) \tag{6.8}$$

This is also known as the *first moment* of the random variable X. Similarly, the n^{th} *moment* of a continuous random variable X is given by:

$$E[X^n] \triangleq \overline{X^n} \triangleq \int_{-\infty}^{\infty} x^n f_X(x) \mathrm{d}x \tag{6.9}$$

with the obvious discrete corollary. Furthermore, the n^{th} *central moment* of a continuous random variable is given as follows:

$$\overline{(X - \overline{X})^n} \triangleq \int_{-\infty}^{\infty} (x - \overline{X})^n f_X(x) \mathrm{d}x \tag{6.10}$$

The second central moment will is used extensively and is referred to as the *variance*. The notation for variance is given by:

$$\sigma_N^2 \triangleq \overline{(X - \overline{X})^2} \triangleq \overline{X^2} - (\overline{X})^2 \tag{6.11}$$

Relationship Between Exponential and Poisson Distributions

The following is based on [36]. While the Poisson distribution describes the number of appearances in a given time interval, it cannot be simulated directly however. To do that we must use the exponential distribution. This distribution is used to describe time between events. It is related to the Poisson distribution as follows.

The cumulative distribution function for exponential density is given by

$$F_X(x) = 1 - e^{-\lambda x} \tag{6.12}$$

The probability density function for the exponential distribution is defined as follow:

$$f_X(x) = \begin{cases} \lambda e^{-\lambda x}, & 0 \leq x < \infty \\ 0, & x < 0 \end{cases} \tag{6.13}$$

Define $X_1, X_2, \ldots$ as a sequence of independent exponentially distributed random variables with parameter λ. X_i is the time between requests. Given an interval of time $[0, t)$, let Y denote the random variable that measures the number of requests in that time interval and let $S_n = X_1 + X_2 + \ldots + X_n$. This means that the probability that n requests occur in the time interval is given by

$$P(Y = n) = P(S_n \leq t, S_{n+1} > t)$$

The density function for S_n is the convolution of the density functions for each exponential density.

$$f_{S_n}(x) = (f_{X_1} * f_{X_2} * \ldots * f_{X_n}) = (\lambda e^{-\lambda x} * \lambda e^{-\lambda x} * \ldots * \lambda e^{-\lambda x})$$

In the following we derive the density function. We will start by calculating the convolution of two exponential density functions f_{X_1} and f_{X_2}.

$$\begin{aligned} f_{S_2} &= \int_0^z \lambda e^{-\lambda(z-y)} \lambda e^{-\lambda y} \mathrm{dy} \\ &= \int_0^z \lambda^2 e^{-\lambda z} \mathrm{dy} \\ &= \lambda^2 z e^{-\lambda z} \end{aligned}$$

Next we will take the convolution of f_{S_2} and f_{X_3}

$$\begin{aligned} f_{S_2} &= \int_0^z \lambda^2 (z-y) e^{-\lambda(z-y)} \lambda e^{-\lambda y} \mathrm{dy} \\ &= \int_0^z \lambda^3 z e^{-\lambda z} - \lambda^3 y e^{-\lambda z} \mathrm{dy} \\ &= \lambda^3 z e^{-\lambda z} y - \frac{1}{2} \lambda^3 e^{-\lambda z} y^2 \Big|_0^z \mathrm{dy} \\ &= \frac{\lambda^3 z^2 e^{-\lambda z}}{2} \end{aligned}$$

Continuing in a similar manner, it can be shown that the general density function for S_n is given by

$$f_{S_n} = \begin{cases} \lambda \frac{(\lambda x)^{n-1} e^{-\lambda x}}{(n-1)!} & x > 0 \\ 0 & x \leq 0 \end{cases}$$

The cumulative distribution function is given by

$$F_n(x) = \begin{cases} 1 - e^{-\lambda x} \left(1 + \frac{\lambda x}{1!} + \ldots + \frac{(\lambda x)^{n-1}}{(n-1)!}\right) & x > 0 \\ 0 & x \leq 0 \end{cases}$$

The probability $P(Y = n)$ becomes

$$\begin{aligned} P(Y = n) &= P(S_n \leq t) - P(S_{n+1} \leq t) \\ &= F_n(x) - F_{n+1}(x) \\ &= e^{-\lambda x} \frac{(\lambda x)^n}{n!} \end{aligned}$$

as every term but the $n + 1$ term cancels out. This is identical to the poisson distribution. This means that if the time between requests follows an exponential density, the overall number of requests in a time interval follows the poisson distribution.

Finally a random variable that satisfies the exponential density is required. We will use the random variable

$$Y = -\frac{1}{\lambda} \ln(rnd) \tag{6.14}$$

Theorem 6.3.1 *Let X be a continuous random variable, and suppose that $\phi(x)$ is a strictly increasing function on the range of X. Define $Y = \phi(X)$. Suppose that X and Y have cumulative distribution functions F_X and F_Y respectively. Then these functions are related by*

$$F_Y(y) = F_X(\phi^{-1}(y))$$

If $\phi(x)$ is strictly decreasing on the range of X, then

$$F_Y(y) = 1 - F_X(\phi^{-1}(y))$$

Corollary 6.3.1 *If $F(y)$ is a given cumulative distribution function that is strictly increasing when $0 < F(y) < 1$ and U is a random variable with uniform distribution on $[0, 1]$, then*

$$Y = F^{-1}(U) \tag{6.15}$$

has the cumulative distribution $F(y)$

This means that setting $Y = F^{-1}(rnd)$ simulates a random variable with the cumulative distribution F.

The random variable Y given by (6.14) satisfies Corollary 6.3.1. So, to simulate a Poisson process on a time interval $[0, t]$ we start be calculating a series of Y values, $y_1, y_2, \ldots, y_m$ using equation 6.14, representing the time between each successive request. Let $x_1, x_2, \ldots, x_m$ denote the times of each request. Each request time is given by the formula

$$x_n = \sum_{i=1}^{n} y_i \tag{6.16}$$

Thus, the random variable X follows a Poisson distribution.

Appendix II: TSP Solutions Methods

Traveling Salesman Problem

Much of the information in this section comes from the excellent book on the traveling salesman problem [37]. The traveling salesman problem has been around in some form for quite a while. One of the reasons for this is the generality of the problem. It has seen applications in logistics, genetics, manufacturing, telecommunications, and neuroscience to name a few. It is one of the few mathematical problems that has made its way into popular culture. Despite its seeming simplicity, a general solution has yet to be found. A simple definition of the traveling salesman problem is to find the cheapest way of visiting a set of cities and return to the starting point. This ordering of cities is called a *tour* or *circuit*. One of the first appearances of tours and circuits in mathematical literature is in a 1757 paper by Leonhard Euler. This paper concerns a solution to the *knight's tour* problem in chess, that is, finding a sequence of knight's moves that will take the piece from a starting square on a chessboard, through every other square exactly once and returning to the starting square. One solution to the knight's tour is shown in Figure 6.1.

42	57	44	9	40	21	46	7
55	10	41	58	45	8	39	20
12	43	56	61	22	59	6	47
63	54	11	30	25	28	19	38
32	13	62	27	60	23	48	5
53	64	31	24	29	26	37	18
14	33	2	51	16	35	4	49
1	52	15	34	3	50	17	36

Figure 6.1: Knight's tour discovered by Euler.

Variations of the tour finding problem have been studied for centuries, although without much mathematical rigor due to the rapid increase in complexity with only a small increase in problem size.

Modern studies of the traveling salesman problem most likely began in the 1930's at Princeton University. This is likely where the name traveling salesman problem first originated. Because of the inherently geometric nature of the problem, the initial work on the problem was done from this perspective. One can look at a tour and very quickly determine if the tour is close to optimal or not by visual inspection. The ability to visualize tours and manipulate them by hand has certainly contributed to the popularity of the problem. A valid tour in the traveling salesman problem is known as a *Hamiltonian circuit*. This term stems from Hamilton's investigation of the dodecahedron, the 12-sided platonic solid [37].

One of the first geometric investigations of the traveling salesman problem took place in the 1940's by Mahalanobis [38] in connection with his studies on farmland surveys. Based on the observation that given a set of n cities with coordinates chosen randomly and uniformly from the unit square, there seems to be a statistical relationship between the number of cities and the length of the optimal tours. Mahalanobis [38] makes the argument that the tour length should grow with a ratio of approximately $\sqrt{n}$. This observation lead to the result of Beardwood, Halton, and Hammersley [39]. This result states that with a probability of 1, as $n \to \infty$, the optimal tour length divided by $\sqrt{n}$ will approach a constant value β. They show analytically that $0.6249 \leq \beta \leq 0.9204$. Modern results such as [40] and [41] place the value of β at $\approx 0.7124 \pm 0.0002$ and $\approx 0.70787 \pm 0.0008$ respectively. These values were attained experimentally. Goemans and Bertsimas [42] showed that β provides an good estimate of the Held-Karp lower bound. This is of interest, because the Held-Karp lower bound is an excellent measure of the optimal tour length and can be used to compare different solutions methods without knowing the optimal tour. However, the Held-Karp lower bound can be computationally expensive for large numbers of cities, so an accurate estimate is extremely valuable for quickly determining bounds on solutions.

Another geometric result was shown by Flood [30], and is the basis for the most successful TSP heuristic to date known as the Lin-Kernighan algorithm. He shows that any two edges in a tour that cross can be removed and replaced by another two edges reducing the overall tour length while still satisfying the constraint that the tour form a Hamiltonian cycle.

Problem Formulation

Next we will give the two most commonly used definitions of the traveling salesman problem as an introduction into a more in-depth discussions of the currently used methods for solving the traveling salesman problem. The two common mathematical formulations of the traveling salesman problem are a graph theoretical model and model based on linear programming. We will start with the graph theoretic theoretic formulation.

Definition 6.3.4 ([43]) *Let $G = (V, E)$ be a graph (directed or undirected) and $\mathcal{F}$ be the family of all Hamiltonian cycles (tours) in G. For each edge $e \in E$ a cost (weight) c_e is prescribed. Then the* traveling salesman problem *is to find a tour (Hamiltonian cycle) in $\mathcal{F}$ such that the sum of the costs of edges of the tour is as small as possible.*

The next definition of the traveling salesman problem derived from the recognition that for a TSP with n cities, the edges and edge costs can be represented by $n \times n$ matrices.

Definition 6.3.5 (Traveling Salesman Problem) *Given a set of n cities denoted by V*

$$\min \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} x_{ij} \quad \textit{where} \quad x_{ij} = \begin{cases} 1 & \textit{if } i \to j \textit{ is in the tour} \\ 0 & \textit{otherwise} \end{cases} \tag{6.17}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n \tag{6.18}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n \tag{6.19}$$

$$\sum_{i \in K} \sum_{j \in K} x_{ij} \leq |K| - 1 \qquad \forall K \subset \{1, \ldots, n\}, 2 \leq |K| \leq n - 1 \tag{6.20}$$

where K is a nonempty proper subset of the set of cities V

The matrix x is the solution vector. Constraints 6.18 and 6.19 ensure that each vertex appears only once in the tour, while constraint 6.20 is known as the subtour elimination constraint. This constraint ensures that the solution is a Hamiltonian circuit, not two or more disjoint cycles. This problem formulation is identical to linear programming optimization with the exception of the subtour elimination constraint and the integer restriction on the decision variable x_{ij}.

Exact Solution Methods

The most common method for finding exact solutions to the TSP is branch and bound. There are two branch and bound formulations based on the relaxation of two mathematical characteristics of a Hamiltonian cycle [41].

i) The graph is connected. Any vertex can be reached by starting at any other vertex

ii) Every vertex has degree 2. There are two edges incident to each vertex

The first relaxation is known as the assignment problem. In this formulation the constraint that the graph must be connected is relaxed while every vertex must have exactly two edges incident is held. This allows only the formation of cycles, but there may exist several disjoint cycles in the solution. The assignment problem can be solved in polynomial time by the Hungarian Algorithm, named after the Hungarian mathematician Kuhn [34].

The other relaxation is the minimum 1-tree relaxation. In this case, the constraint that every vertex has exactly two edges incident is relaxed while the constraint that the graph be connected is held. This relaxed problem can also be solved in polynomial time by minimal spanning tree algorithms such as the ones by Kruskal [44] or Prim [45].

If a solution to one of these relaxations satisfies both properties, it is the optimal tour.

These methods are dual in the sense that they possess properties that are complementary with respect to the properties of the TSP [46].

Assignment Problem Relaxation

The assignment problem branch and bound method begins with a graph $G = (V, E, C)$ with n vertices and a corresponding cost matrix $C = [c_{ij}]$ where $c_{ii} = \infty, \forall i \in \{1, \ldots, n\}$. The condition $c_{ii} = \infty$ removes all cycles of a single vertex. This condition is not essential to the formulation, but if added will reduce the computation time. Initially, solving the assignment problem on the matrix C will yielding m circuits where $1 \leq m \leq n - 1$ [47]. If $m = 1$ then the solution to the assignment problem is the solution to the corresponding traveling salesman problem.

If $m > 1$ as is often the case, one circuit of arbitrary degree k given by $S = (v_1, v_2, \ldots, v_k, v_1)$ is chosen. In a valid TSP solution, this circuit must not exist. In addition at least one arc must have one endpoint in the set of vertices S and the other endpoint in the remaining vertices $\overline{S} = V \setminus S$. The existence of an arc from S to $\overline{S}$ guarantees that the subproblems are disjoint, and the solutions containing circuit S are eliminated [48]. Each vertex in S will become a subproblem $P_1, P_2, \ldots, P_k$.

For an individual subproblem P_i, this condition is simply removing the edges from vertex v_i to all the other vertices in S. This is done practically by setting $c_{ij} = \infty, \forall v_j \in S$. This eliminates the possibility that v_i can connect with any vertex in S, leaving only the vertices $\overline{S}$ to connect with. The assignment problem is then solved for each subproblem. This constitutes the branching rule implemented.

The bounds come from the total cost of the assignment. The elimination of each branch (or node) is determined by the current upper bound, given by the cost of the current best solution and the lower bound on each node, given by the cost of the solution to the assignment problem at each node. If the lower bound on the cost at a node is greater that the upper bound on the solution value, the node is eliminated. This is because each subproblem of that node will have a cost greater than the best solution so far.

Minimum 1-Tree Relaxation (Subgradient Optimization) (Langrangean Relaxation)

The basic premise of this method is to find a series of minimum 1-trees where the degree of all vertices approach 2. A minimum one tree is defined as the minimum spanning tree found for the vertices $v_2, v_3, \ldots, v_n$ and the two lowest cost edges incident to vertex v_1. The main emphasis in this method it finding the best lower bound on each node, then branching.

To bound the problem, the Held-Karp lower bound is the bound most commonly used. This bound is found by an iterative process. Let the TSP be defined as an $n \times n$ graph $G = (V, E, C)$. Let the cost of the optimal tour of G be denoted as c^*. The object is to find a lower bound $w(\pi) \leq c^*$ as close to c^* as possible. This is done by modifying the set of edge costs c_{ij} by a set of evolving weights $[\pi_1, \pi_2, \ldots, \pi_n]$ on each vertex that increases the bound. These vertex weights modify the cost matrix in the following manner.

$$\overline{c}_{ij} = c_{ij} + \pi_i + \pi_j \tag{6.21}$$

The new minimum 1-tree found for the graph $G(V, E, \overline{C})$ will, in general, be different than the 1-tree for the original graph and of higher cost, however, the ordering of the tour lengths for the set of all possible tours will not be affected because the transformation adds $2\sum \pi_i$ to the cost of every tour [41]. The weights are defined by $\pi_i = d_i - 2$ where d_i is the degree of vertex i. This can be repeated multiple times.

We begin by denoting the cost of a 1-tree or tour T with respect to a cost matrix c_{ij} by $L(c_{ij}, T)$. Let X be the set of tours on G and Y be the set of 1-trees on G. The set of tours is the subset of 1-trees with vertices of degree 2 which implies that $X \subseteq Y$ and

$$\min_{T \in Y} L(\overline{c}_{ij}, T) \leq \min_{T \in X} L(\overline{c}_{ij}, T) \tag{6.22}$$

This is the starting point for determining an expression for the lower bound $w(\pi)$. Next we define the relationship between the cost of the minimum 1-tree before and after the transformation of the

cost matrix. Since $\overline{c}_{ij} = c_{ij} + \pi_i + \pi_j$

$$L(\overline{c}_{ij}, T) = L(c_{ij}, T) + \sum_{i=1}^{n} \pi_i d_i^T \tag{6.23}$$

where d_i^T is the degree of the i^{th} vertex of the 1-tree T. If T is a tour, then $d_i^T = 2$ for all vertices and the equation becomes

$$L(\overline{c}_{ij}, T) = L(c_{ij}, T) + \sum_{i=1}^{n} 2\pi_i \tag{6.24}$$

Let $L(\overline{c}_{ij}, T^*)$ denote the cost of the minimum tour for the modified cost matrix, and let $c^* = L(c_{ij}, T^*)$ denote the cost of the minimum tour of the original problem. Substituting (6.24) for the RHS of (6.22), we have the following

$$\min_{T \in Y} L(\overline{c}_{ij}, T) \leq \min_{T \in X} L(\overline{c}_{ij}, T^*) \tag{6.25}$$

$$\min_{T \in Y} L(\overline{c}_{ij}, T) \leq L(c_{ij}, T^*) + \sum_{i=1}^{n} 2\pi_i \tag{6.26}$$

$$\min_{T \in Y} L(\overline{c}_{ij}, T) \leq c^* + \sum_{i=1}^{n} 2\pi_i \tag{6.27}$$

Next denote the cost of any 1-tree with respect to the original problem as $c_T = L(c_{ij}, T)$. Substituting into (6.23) yields

$$L(\overline{c}_{ij}, T) = c_T + \sum_{i=1}^{n} \pi_i d_i^T \tag{6.28}$$

Substituting (6.28) into (6.27) yields

$$\min_{T \in Y} \left\{ c_T + \sum_{i=1}^{n} \pi_i d_i^T \right\} \leq c^* + \sum_{i=1}^{n} 2\pi_i \tag{6.29}$$

Hence for any set π

$$\min_{T \in Y} \left\{ c_T + \sum_{i=1}^{n} \pi_i (d_i^T - 2) \right\} \leq c^* \tag{6.30}$$

This leads to the definition of the lower bound for the optimal tour as

$$w(\pi) = \min_{T \in Y} \left\{ c_T + \sum_{i=1}^{n} \pi_i (d_i^T - 2) \right\} \tag{6.31}$$

Finally we define the *Held-Karp lower bound* as

$$HK = \max_{\pi} w(\pi) \tag{6.32}$$

In general HK is not equal to the length of the minimum tour but is usually very close [41].

After the *Held-Karp lower bound* has been calculated branching can take place. This is done by choosing a vertex with $d_i^T \geq 3$. The number of nodes in the branching is equal to d_i^T. For each node, one of the edges incident to the vertex is removed and the *Held-Karp lower bound* is then

calculated for each of these nodes. If this lower bound is greater than the upper bound (the cost of the best known solution), that node is discarded.

Dynamic Programming

The next class of TSP algorithms arises from Richard Bellman's [49] theory of *dynamic programming*. The foundation of this algorithm is the fact that after a number of cities has been traversed on the optimal tour, the path through the remaining cities must be optimal. Thus, using a list of all possible minimum-cost paths through subsets of k-cities, we can build tours step by step. In [50], Held and Karp show that the dynamic programming algorithm can solve an n-city TSP in at most $\mathcal{O}(n^2 2^n)$ time, which has the honor of being the best analytic time complexity for an algorithm that solves all instances of the TSP. While this is still a weak bound, it is much stronger than the time to enumerate all tours. Despite this analytic bound, the practical running time restricts the use of this algorithm because of the amount of data that needs to be stored and processed to list all of the minimum cost paths through k cities, while other methods that do not have an analytic bound show better solution time in practice.

Branch and Cut

Modern TSP solvers use a hybrid of branch and bound methods and cutting plane methods known as *branch and cut*. The first successful application of this method to the traveling salesman problem was by Padberg and Rinaldi [51] who coined the name branch and cut [37]. Their code solved 42 instances of the TSP with sizes from 48 to 2,392 cities. The main idea of branch and cut is to implement a linear programming relaxation, that allows nodes to be trimmed without resorting to branching, thereby reducing the number of nodes to be checked. The current record holder is the *Concorde* code by Applegate et. al. This code employs the most modern version of the branch and cut method in a program consisting of roughly 130,000 line of code [37].

Table 6.1: Solutions of TSP instances with Concorde.

1992	Concorde	3,038 cities	pcb3038
1993	Concorde	4,461 cities	fnl4461
1994	Concorde	7,397 cities	pla7397
1998	Concorde	13,509 cities	usa13509
2001	Concorde	15,112 cities	d15112
2004	Concorde	24,978 cities	sw24978
2004	Concorde with Domino-Parity	33,810 cities	pla33810
2006	Concorde with Domino-Parity	85,900 cities	pla85900

Table 6.1 shows the record breaking solutions by Concorde. These are all instances from TSPLIB, a library of standard TSP problems maintained by the University of Heidelberg [52].

Approximate Solution Methods

While exact solutions to the TSP are of interest, they are impractical to implement for a real-time system. The ability to find the exact solution does provide a baseline for the comparison of different TSP heuristics. The majority of these methods are suitable for real-time implementation.

Many combinatorial optimization problems can be expressed as 'find from a set E a subset T that satisfies some criterion C and minimizes an objective function f.' The traveling salesman problem

is no exception. Given a graph $G = (V, E)$ find a subset of the edges T that is a Hamiltonian tour (condition C) and minimizes the tour length (objective function f). This is the basis for heuristically solving combinatorial optimization problems [28]:

i) Generate an initial random or pseudo-random solution T that satisfies condition C.

ii) Attempt to find a transformation T' of the initial solution T that improves the cost

iii) If a solution is found that satisfies $f(T') < f(T)$, then replace T with T' and repeat step *ii*).

iv) If no improving solution is found, then T is a locally optimal solution.

In this method there are two types of heuristics, tour creation heuristics and local improvement heuristics.

Tour Construction Heuristics

The most commonly used tour construction heuristic is the Nearest Neighbor (NN) heuristic. The algorithm starts at a random city then selects the nearest neighbor not in the tour as the next definition. Denoting the length of the optimal tour as L_{opt} and the length of a NN tour L_{NN}, we are guaranteed a bound of

$$\frac{L_{NN}}{L_{opt}} \leq 0.5(\lceil \log_2 N \rceil + 1) \tag{6.33}$$

which grows with the size of the problem.

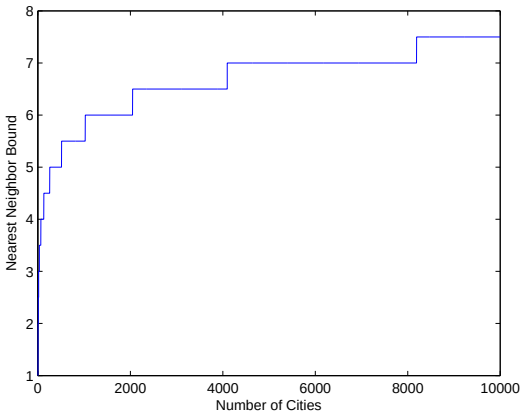


Figure 6.2: Upper bound on the initial nearest neighbor tour quality.

The best tour construction heuristic is the Christofides heuristic. It guarantees a bound of $L_C/L_{opt} \leq 1.5$ although at a significant increase in running time. [29]

One tour creation heuristic that is not used in practice very often is the Nearest Insertion (NI) heuristic. This heuristic guarantees a bound of $L_{NI}/L_{opt} \leq 2$ which is not a function of problem size. The NI heuristic starts with three random nodes, then inserts vertices one at a time into the edge of the existing tour that increases the overall cost the least.

Local Improvement Heuristics

Local improvement heuristics are used after the an initial solution has been found to transform that solution into a solution with a lower cost. For the traveling salesman problem, given the graph

$G = (V, E)$, there exists a set of feasible solution $\mathbf{S} \subset E$. In that set of feasible solutions, there exists one solution $\hat{T}$ that is the global minimum. Next, given a general feasible solution F, and a transformation F' of the feasible solution that satisfies $f(F') < f(F)$, then there is a set of locally minimum solutions $\mathbf{F} \subset \mathbf{S}$. If the transformation is applied to the set of feasible solutions, it will by definition transform $\mathbf{S} \rightarrow \mathbf{F}$. The better the transformation or heuristic, the smaller the set of locally optimal solutions is. A heuristic where $\hat{T} \equiv \mathbf{F}$ will solve the TSP for every initial condition. However, there does not exist any heuristic for which this condition is true.

The main transformation that is used on the TSP is an interchange method. Given a graph $G = (V, E)$ of size n and a tour T on that graph with the optimal tour denoted by $\hat{T}$, there are $0 \leq k \leq n$ edges that are different in T and $\hat{T}$. In practice finding these k edges or even the size of k is impossible. However, by fixing k the tour can be improved locally. To do this we exchange k elements from T with k elements from $E \setminus T$ such that the resulting solution T' is a tour and has less cost than T. These moves are called *k-opt* moves, k edges are removed from the tour and then the remaining chains are connected by k different edges in a manner to reduce the overall tour cost.

There are three local improvement heuristics currently used and have been in use for over thirty years. These are the *2-opt*, *3-opt*, and *Lin-Kernighan* heuristics. Of these heuristics, the *Lin-Kernighan* is a generalization of the first two and provides better results and is the most widely implemented tour improvement heuristic. The LK algorithm attempts to find a series of sequential *2-opt* moves that improve the tour cost.

Lin-Kernighan The most successful approach to tour finding was developed by Lin and Kernighan in 1973 [28]. This work is based on papers by Flood [30] and Croes [53]. Flood removes intersections in a tour by noting that in an n-city tour $v_0, \ldots, v_{n-1}$, for each $0 \leq p < q < n$ we have

$$c_{v_{p-1}v_p} + c_{v_q v_{q+1}} \leq c_{v_{p-1}c_{v_q}} + c_{v_p v_{q+1}} \tag{6.34}$$

where subscript are modulo n. Graphically, any pair of edges (v_{p-1}, v_p) and (v_q, v_{q+1}) that violate 6.34 are an *intersecting pair* and any tour with an intersecting pair is not optimal.

Flood's method was to find an initial tour, then iteratively remove all intersecting pairs.

Croe's described a procedure for finding an intersectionless tour by a sequence of inversions. A section of the tour is removed, flipped then inserted back into the tour in a manner that removes intersections. Lin and Kernighan [28] defined a $k-optimal$ tour as a tour in which it is not possible to reduce the tour cost by replacing any of its k edges.

To describe an algorithm we need some additional notation. Given an oriented tour, let *next(v)* denote the vertex immediately following v in the tour. Likewise, let *prev(v)* denote the vertex immediately previous to v in the tour. Let *c(v1,v2)* denote the cost of the edge from $v1$ to $v2$. Let *(base,next(base))* be the edge we are trying to improve and *(probe,next(probe))* is the edge that that for potential improvement.

As shown in Figure 6.3, a *2-opt* move would improve the solution if and only if

$$\begin{gathered} c(base, next(base)) + c(probe, next(probe)) > \\ c(next(base), next(probe)) + c(base, probe) \end{gathered} \tag{6.35}$$

Lin-Kernighan, however, uses a greedy approach and only requires that

$$c(base, next(base)) - c(next(base), next(probe)) > 0 \tag{6.36}$$

Restricting the move to the interchange of two edges reduces LK to the *2-opt* heuristic. LK allows a series of moves, where the cost of an individual move may increase the cost of the solution,

as long as the overall tour length is decreased. This greatly expands the class of allowable moves and yields much better solutions. This is variable *k-opt* where a *k-opt* move is built from a series of sequential *2-opt* moves. This is done by defining a quantity *delta* which is the amount of local improvement that has been obtained at the current point in the sequence of *2-opt* moves. *delta* starts at 0 and at each step is incremented by

$$\begin{array}{c} c(base, next(base)) - c(next(base), next(probe)) \\ +c(probe, next(probe)) - c(probe, base) \end{array} \tag{6.37}$$

which is the amount of change in the tour length at that step. By considering vertices which satisfy

$$delta + c(base(next(base)) - c(next(base), next(probe)) > 0 \tag{6.38}$$

we allow individual moves that increase the cost if we might be able to later realize an improvement [37]. Notice that if the term $c(next(base), next(probe))$ is small enough the vertex *probe* is promising since the other two terms do not depend on probe. This condition can be checked very quickly.

Just choosing the first promising edge is shortsighted, we want to use the edge that maximizes the gain from the 2-opt move. Using a term from inequality 6.35 choose the edge that maximizes

$$c(probe, next(probe)) - c(next(base), next(probe)) \tag{6.39}$$

This is repeated until the solution converges to a locally optimal solution.

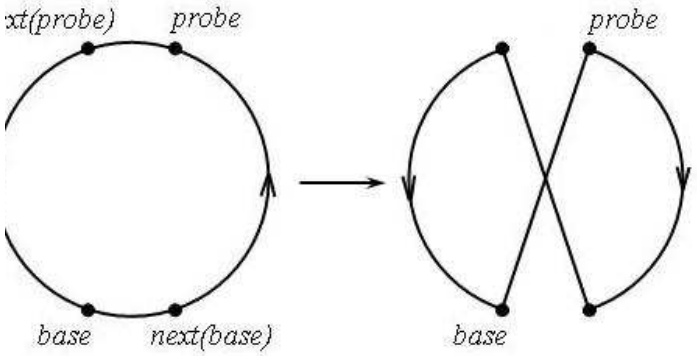


Figure 6.3: A 2-opt move

Vita

Alex Melin was born in Glens Falls, New York, on February 24, 1979 to Michael and Sharon Melin. After graduating in 1997 from Carthage High School in Carthage Missouri, he attended the University of Missouri - Columbia where he received his Bachelor of Science degree in 2001 and a Master of Science Degree in 2003 in Mechanical Engineering. While at the University of Missouri, he was a member of the Mizzou Solar Car Team. He was involved in manufacturing of SunTiger IV which placed 6^{th} in Sunrayce '99, 3^{rd} in Formula Sun in 2000, and 11^{th} in the 2001 American Solar Challenge. After the 2001 ASC, he became the Head of Design for SunTiger V. In addition to managing the mechanical, electrical, telemetry, and strategy design for the team, he designed a custom semi-monocoque chassis and body which reduced the chassis weight to 2.4 lbs, a 93% reduction from the previous chassis, and reduced the overall car weight to 69% of the previous car. SunTiger V placed 6^{th} in the 2003 American Solar Challenge. While in graduate school, Alex participated for five summers in the Space Scholars program, an internship with the Air Force Research Lab, Space Vehicles Directorate at Kirtland Air Force Base, in Albuquerque, New Mexico. During this internship, he performed research involving autonomous scheduling for satellites which evolved into the basis for this dissertation research into bandwidth-on-demand for laser communications satellites. In 2004 Alex moved to Knoxville Tennessee to pursue his Doctor of Philosophy in Mechanical Engineering with an emphasis on control theory. Outside of school, Alex is an avid musician, playing drumset and hand drums. He also dabbles in photography.