

DETERMINING TRUNK RESERVATION PARAMETERS IN A
NON-SYMMETRIC TELECOMMUNICATIONS NETWORK
WITH NON-UNIFORM DEMAND

by

Thaddeus Kim Teck Sim

Bachelor of Science (Mathematics and Finance), University of Alberta, 1998
Bachelor of Commerce (Operations Management), University of Alberta, 2000

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE (BUSINESS ADMINISTRATION)

in

THE FACULTY OF GRADUATE STUDIES

~~Faculty~~
(Department of Commerce and Business Administration)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

December 2001

© Thaddeus Kim Teck Sim, 2001

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of COMMERCE AND BUSINESS ADMINISTRATION

The University of British Columbia
Vancouver, Canada

Date DECEMBER 13, 2001

Abstract

When a telephone call enters a fully-connected telecommunications network, the network first attempts to connect the call via a direct path. If this is not possible, the network attempts to connect it through a combination of two or more trunk groups. This is called a tandem path. When the network reaches operational capacity, the use of tandem paths is undesirable because the trunk groups in the path could be used to connect more than one telephone call.

Trunk reservation is the partitioning of trunk groups to allow only direct calls to connect on the reserved trunks. This reduces the use of tandem paths when the network is at capacity, thus increasing the probability of connecting more calls on the more efficient direct path. Past studies in this area have focused on using uniform trunk reservation parameters in symmetric networks with uniform demand. In this thesis, we attempt to determine non-uniform trunk specific reservation parameters in a non-symmetric network with non-uniform demand using fixed-point approximations of a birth-death process and simulated annealing. An application of this study to TELUS' Edmonton network is discussed.

Table of Contents

ABSTRACT.....	ii
TABLE OF CONTENTS.....	iii
LIST OF TABLES	v
LIST OF FIGURES	vi
ACKNOWLEDGEMENTS.....	vii
CHAPTER 1: INTRODUCTION.....	1
1.1 TELUS and the Communications Industry in Canada	1
1.2 Operations of a Telecommunication Network.....	1
1.3 Circuit and Packet-Switched Networks	3
1.4 TELUS and the Centre for Operations Excellence	4
1.5 Trunk Reservation.....	5
1.6 Applying Trunk Reservation in TELUS' Edmonton Network.....	8
CHAPTER 2: LITERATURE REVIEW AND PRELIMINARIES	10
2.1 Introduction	10
2.2 Literature Review on Trunk Reservation.....	10
2.3 Combinatorial Optimization Heuristics	12
2.3.1 Tabu Search	12
2.3.2 Simulated Annealing	13
2.3.3 Genetic Algorithm.....	14
CHAPTER 3: MODELING THE TRUNK RESERVATION NETWORK.....	16
3.1 Introduction	16
3.2 A Birth and Death Process	16
3.3 Stationary Probability Distributions	18
3.4 Modeling the Arrival and Service Rates	20
3.5 Calculating the Non-First-Routed Traffic Arrival Rates.....	21
3.6 Independence of Paths in a Route.....	24

CHAPTER 4: PROBLEM FORMULATION AND SOLUTION METHODOLOGY	26
4.1 Introduction	26
4.2 Heuristics, Mathematical Optimization and Simulation	26
4.3 Simulated Annealing: Problem Specific Details	28
4.3.1 Simulated Annealing Algorithm	28
4.3.2 Objective Function, $f(S)$	28
4.3.3 Neighbourhood	29
4.3.4 Initial Temperature, T	29
4.3.5 Length of Loop, L	30
4.3.6 Temperature Reduction Rate, r	31
4.3.7 Stopping Criterion	31
CHAPTER 5: RESULTS	33
5.1 Introduction	33
5.2 Simulated Annealing Results	33
5.3 Simulation Results	36
5.4 Process to Determine the Final Optimal Reservation Parameters	41
5.4.1 Initial Temperature Acceptance Rate	42
5.4.2 Reduction Rate	44
5.4.3 Acceptance Rate	45
5.4.4 Final Reservation Parameters	46
CHAPTER 6: CONCLUSIONS	47
6.1 Study Objectives	47
6.2 Future Studies	47
APPENDIX 1: FINAL OPTIMAL TRUNK RESERVATION PARAMETERS	49
REFERENCES	52

List of Tables

Table 1: Example of a routing table	2
Table 2: Total overflow levels for networks with and without trunk reservation from the fixed-point approximation and simulation models	39
Table 3: Average and standard deviations of the spreads for different initial temperature acceptance rates.....	43
Table 4: Average and standard deviations of the spreads for different reduction rates.....	44
Table 5: Averages and standard deviations of the spreads for different acceptance rates.....	45

List of Figures

Figure 1: A three-switch network without trunk reservation	5
Figure 2: Routing calls through a three-switch network without trunk reservation.....	6
Figure 3: A three-switch network with trunk reservation.....	7
Figure 4: Routing calls through a three-switch network with trunk reservation	7
Figure 5: Distribution of trunk group capacities in the Edmonton network	8
Figure 6: Transition diagram of a birth-death process to model a single trunk group.....	16
Figure 7: An example to illustrate the calculation of the total arrival rate onto trunk group CD.....	22
Figure 8: Evolution of the total overflow level for a simulated annealing process	34
Figure 9: Distribution of the suggested number of reserved trunks for the trunk group between switches 6 and 9.....	35
Figure 10: Distribution of the suggested number of reserved trunks for the trunk group between switches 7 and 8	35
Figure 11: Total overflow levels for networks with and without trunk reservation at various load scale factors.....	37
Figure 12: P-values of two-sample t-tests on the means of the total overflow levels of the no-reservation and optimal reservation networks	38
Figure 13: Average CCS per trunk for networks with and without trunk reservation at various load scale factors.....	40
Figure 14: Percentage of multi-link calls for networks with and without trunk reservation at various load scale factors.....	41
Figure 15: Illustration showing which parameter value is better	42
Figure 16: Plot of the averages and standard deviations of the spreads for different initial temperature acceptance rates.....	43
Figure 17: Plot of the averages and standard deviations of the spreads for different reduction rates	45
Figure 18: Plot of the averages and standard deviations of the spreads for different acceptance rates	46

Acknowledgements

Firstly, many thanks to my parents for their support throughout all the years I have spent in school up to the completion of this Master's program. More importantly, I am thankful for your trust and belief that I know what I am doing when often times I am just as lost.

I am grateful to have Professor Erhan Erkut as a mentor and friend. Your energy and enthusiasm for the field of OR is unparalleled and infectious. You saw in me the ability and potential to do well in this field and never allowed me to rest on my laurels. Thanks for the constant push towards excellence and for introducing me to the world of academia.

Sincere thanks to my advisor Professor Hong Chen for the valuable input and comments on my thesis. It has been a pleasure working with and learning from one of the best professors in the field. Thanks to the Centre for Operations Excellence and TELUS for giving me the opportunity to work on such an interesting and challenging project, and also to the TELUS team at the COE – Steven Kabanuk, Sean Baird, Kelly Chung and Michael Thomas. Finally, to Kevin Tang for being a wonderful sounding-board and friend throughout the entire program. I hope that your hard work and sleepless nights will eventually be appreciated and repaid abundantly.

I especially want to thank the communities of St. John's and St. Mark's Colleges. This is the first time I have lived away from home, and the friendships and support you have offered have made my stay in Vancouver enjoyable and one to remember.

Finally, I would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for their financial support.

Carpe diem!

Chapter 1: Introduction

1.1 TELUS and the Communications Industry in Canada

TELUS Corporation is a leading telecommunications Canadian company providing a full range of communications products and services that connect Canadians to the world. The company generated \$6.4 billion in revenues in 2000 and is the leading service provider in Western Canada (TELUS Investor Fact Sheet 2000). TELUS' three largest local telecommunication networks in Western Canada are the Lower Mainland Vancouver, Calgary and Edmonton networks.

In the early 1990's, the Canadian Radio-Television and Telecommunications Commission (CRTC) introduced deregulation of the telecommunications industry in Canada, which allowed for competition on long-distance voice services (Washburn 1996). With cheaper long-distance services, the amount of telephone calls on telecommunication networks increased. In addition, the explosion of dial-up Internet demand in the late 1990's further increased the demand on the networks.

With the increase in demand for telecommunication services, TELUS anticipates that their networks in Western Canada will be operating at capacity in the near future. Furthermore, TELUS is required to maintain a CRTC mandated minimum grade of service of P.01 – a service level where no more than one out of every 100 telephone calls is blocked. Therefore, TELUS has to make a decision on capital expenditures regarding the operations of their networks in the near future.

1.2 Operations of a Telecommunication Network

A discussion on the components and workings of a telecommunication network is presented here to define some terminology and the roles of the components. In a telecommunication network, there are switches, trunk groups and routing tables. Telephony devices such as telephones, fax machines and modems are connected to switches via trunks or links. Trunk groups, which comprise of many trunks, connect switches to one another. Routing tables direct telephone calls entering the network to

their destinations. A path is a set of distinct trunk groups that form a connection between the origin and destination switches. A route is an ordered collection of paths connecting the same pair of switches, specifying the paths used for routing calls between the pair in the order that seizure of the paths is attempted (Akinpelu 1984). Table 1 shows a portion of a routing table containing the route for the origin-destination pair 1-5. This particular route contains three paths for routing a call. The "Route Index" column in the table shows the order of paths a call will attempt.

Origin	Destination	Route Index	Path
1	5	1	1, 5
1	5	2	1, 4, 5
1	5	3	1, 2, 4, 5

Table 1: Example of a routing table

Consider a call originating from switch 1 with switch 5 as its destination. The first path (route index 1) connects the call directly, that is, going from switch 1 to switch 5 through the trunk group connecting the two switches. The first path for any origin-destination pair is typically the direct path, if the trunk group exists. Calls or traffic on the first path is termed first-routed traffic. If a call is connected via this path, a circuit between the origin and destination switches is formed. If all trunks in that trunk group are occupied, the call is overflowed onto an alternate path, which, in this example, is the second path (route index 2). The second path routes the call from switch 1 to switch 5 via switch 4. Switch 4 is called a tandem switch, and this path is called a tandem path. When either of the trunk groups between switches 1 and 4 or switches 4 and 5 are fully occupied, therefore not allowing a circuit to be formed via this path, the call is then overflowed onto the next alternate path. When there are no more paths in the route for a call to overflow to, the call is blocked and the caller receives a network busy signal. This routing system is called alternate routing.

Franks and Rishel (1973) and Kabanuk (2000) provide a more thorough discussion on the operations of a telecommunications network. In several papers describing the operations of a telecommunication network, the distinction between capacities in a trunk

group and in a switch is emphasized. In this thesis, the capacities of the switches are assumed to be large and hence, are not constraints in our analysis.

1.3 Circuit and Packet-Switched Networks

All three of TELUS' networks in Western Canada operate as circuit-switched networks. When a call is placed and completed in a circuit-switched network, a circuit is used for the entire duration of that call. Advanced technology has appeared on the market where telecommunication networks are operated as packet-switched networks, which is similar to that of the Internet. In a packet-switched network, the voice conversation is broken up into packets and transmitted over the network.

Consider any telephone conversation between two parties. Undoubtedly, there will be periods during the telephone call when both parties are silent. If this call occurred in a packet-switched network, these "silent" periods allow the system to route packets from another conversation through the links in this circuit. This opportunity is not available in a circuit-switched network. Therefore, packet-switched networks offer higher utilization opportunities than circuit-switched networks.

As mentioned in Section 1.1, demand for telecommunication services have risen in the past few years. This upward trend in demand will soon result in TELUS' networks operating at capacity. Rapidly rising Internet dial-up traffic further compounds the problem. Since 1997, dial-up connections in the United States have increased at a compounded growth rate of 50% (CISCO 2000), and a similar pattern can be assumed for Canada. In addition, dial-up Internet calls last longer than voice calls and this is one of the main causes of congestion in the networks since the networks were originally engineered to handle voice calls only.

TELUS has identified three options to maintain the required minimum grade of service in the face of the increasing demand. The first option is to augment the existing networks with more circuit-switched technology. However, this legacy technology is very costly and does not provide "the scalability to handle today's networking demands" (CISCO 2000). The second option is to replace the entire network with packet-switching

technology, yet another costly alternative. Moreover, packet-switched technology is relatively new and there are still some reservations towards using this technology for telecommunication purposes. The final option is to evaluate the existing configurations of the networks and identify opportunities to increase their utilization capacities without any excessive capital expenditures. This option, if successful, will delay the decision to either augment the current networks or replace them with packet-switching technology. This delay will also allow packet-switching technology to mature into a more reliable technology.

1.4 TELUS and the Centre for Operations Excellence

For the past two years, TELUS has collaborated with the Centre for Operations Excellence (COE), a research center within the Faculty of Commerce at the University of British Columbia, to examine the operations of their networks. The purpose of this relationship is to provide TELUS with recommendations regarding the feasibility of increasing the capacities of their networks without any immediate capital expenditure outlays. Two studies – call routing methodologies and time-of-day routing – towards providing the recommendations have been completed.

In the time-of-day routing study, Smith (2000) showed that the levels of utilization on each trunk group differ by its service base and the time of day. Trunk groups connecting business areas experienced higher utilization during business hours and lower utilization during non-business hours. Conversely, trunk groups servicing residential areas experienced higher utilization during evening periods and lower utilization during the day. Therefore, TELUS could service more calls by directing calls through the residential portion of the network during the day and through the business portion during evening periods.

Kabanuk's (2000) call routing study determined optimal routing tables for TELUS' Lower Mainland Vancouver network. Kabanuk provided a methodology based on linear optimization to generate new routing tables to take advantage of under-utilized trunk groups based on different times of the day. Kabanuk also showed that modifying the

structure of the network could increase the carried load (utilization) on the Vancouver network.

Both studies showed that there are indeed opportunities for TELUS to increase the utilization capacities of their networks by changing both the operations and configurations of their networks. This thesis adds to the completed work by proposing that the introduction of a control mechanism in the network called trunk reservation will further increase utilization when the network is at overload levels (when demand is greater than the capacity).

1.5 Trunk Reservation

Figure 1 shows a three-switch network that will illustrate the need for trunk reservation. All trunk groups in the network have capacities of three trunks. The direction of the arrows indicates the path for each origin-destination pair. The two paths for a call going from switch A to B, in order of attempts, are $A \rightarrow B$ and then $A \rightarrow C \rightarrow B$. Currently, the trunk group connecting switches A and B is fully occupied, while two of three trunks in the trunk groups connecting switches A to C and C to B are occupied.

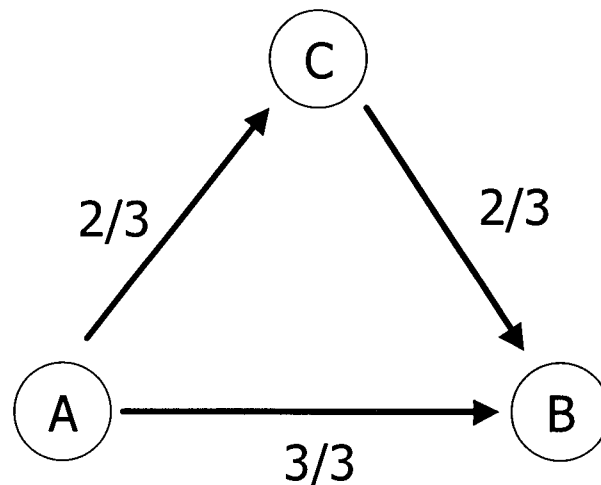


Figure 1: A three-switch network without trunk reservation

Consider the following sequence of calls entering the network: $A \rightarrow B$, $C \rightarrow B$ and $A \rightarrow C$. Since the first call cannot connect via its first path (all trunks occupied on the trunk group $A \rightarrow B$), it is overflowed and connected on the second path with switch C as the

tandem. After routing the first call, all trunk groups in the network are fully occupied as shown in Figure 2. The next two calls, C→B and A→C, are blocked because there are no available trunks to route the call. In this simple example, two out of the three incoming calls are blocked.

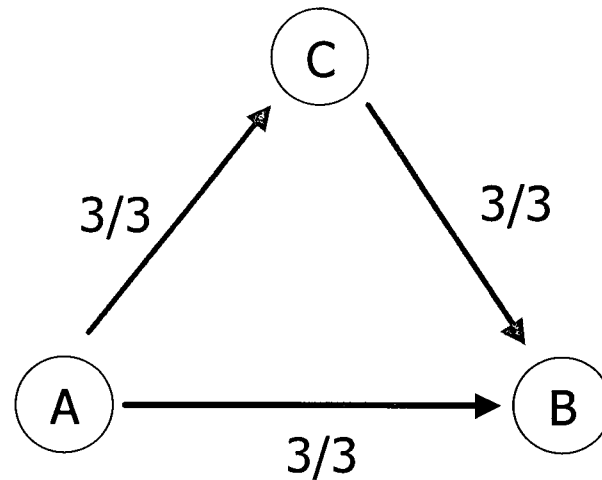


Figure 2: Routing calls through a three-switch network without trunk reservation

In systems with trunk reservation, a portion of a trunk group is reserved for first-routed traffic only (calls connecting on the first path of a route). Consider a call entering the network and is overflowed onto its second path because all trunk groups in its first path are occupied. Assume that in the second path, all the unreserved trunks are occupied while none of the reserved trunks is occupied. The call overflows from the second path because it is not first-routed traffic and hence is not allowed to use the reserved trunks. Reserved trunks are used only when all of the unreserved trunks are occupied.

Figure 3 shows a three-switch network with trunk reservation. All trunk groups have capacities of three trunks each, with one trunk reserved for first-routed traffic only. In the instance of the network shown in Figure 3, all the unreserved trunks are occupied while only the reserved trunk between switches A and B is occupied. Consider again the same three calls entering the network: A→B, C→B and A→C. Since the first call cannot form a circuit on its direct path, it overflows to its next alternate path – the tandem path through switch C. Although there are trunks available to route this call, the network will

not connect the call on this path because the available trunks are reserved for first-routed traffic only. Therefore, the call from A to B is blocked. The next two calls, C→B and A→C, can use the reserved trunks in their respective paths because they are first-routed traffic. The resulting state of the system is shown in Figure 4. In this simple example, only one of three incoming calls is blocked.

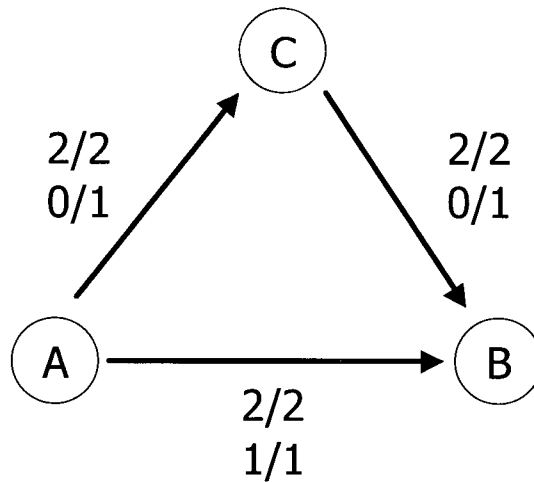


Figure 3: A three-switch network with trunk reservation

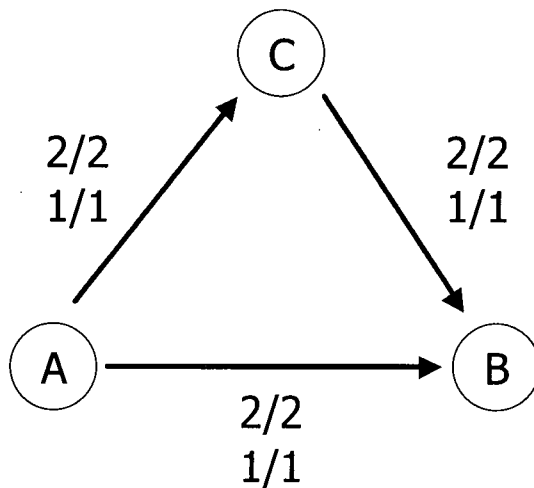


Figure 4: Routing calls through a three-switch network with trunk reservation

Comparing this number of calls blocked to that from the network without trunk reservation, trunk reservation decreased the number of blocked calls by 50%. Although

blockage reductions in networks with trunk reservation will rarely be that high, the potential savings are significant enough to justify the study of its use in a telecommunication network.

1.6 Applying Trunk Reservation in TELUS' Edmonton Network

TELUS' Edmonton local, non-toll network consists of 12 switches. The network is a fully connected network, that is, every switch in the network is connected to all other switches. The trunk group sizes in the network are not uniform, with the number of trunks in a trunk group ranging from 120 to over 1200 trunks as shown in Figure 5. Routing of calls in the Edmonton network is by direct routing only, and therefore, there are opportunities to use these capacities more efficiently through alternate routing because of different time-of-day loads on the network.

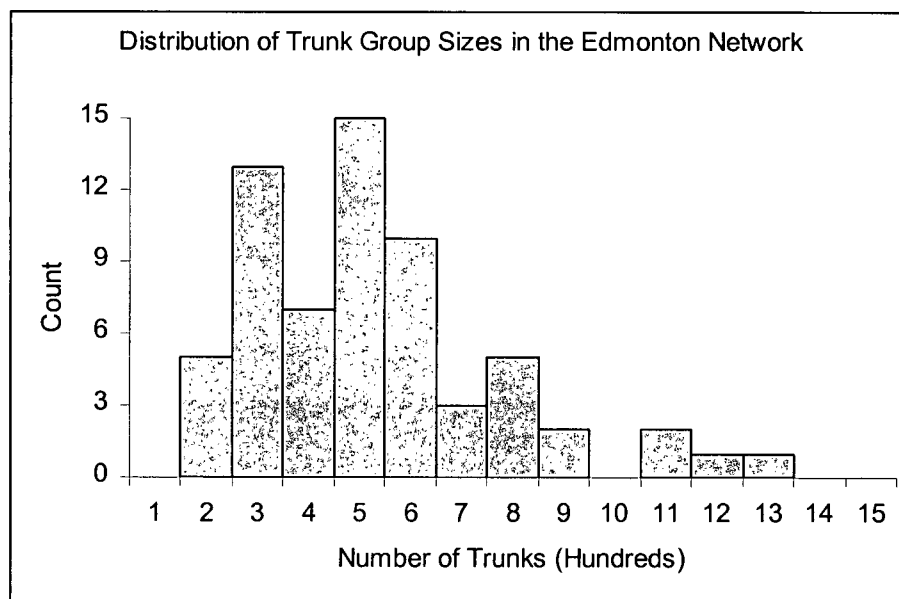


Figure 5: Distribution of trunk group capacities in the Edmonton network

The data for this study is from the one-week period from May 6th to 12th, 2001. The database contains information for all calls occurring in the network during that time-period. This information includes the origin and destination switches for each call, the time a call was made, whether the call was connected or not (busy signals), and if connected, the duration of the phone call.

Most studies on trunk reservation, such as those published by Akinpelu (1984), Gibbons and Kelly (1990) and Krupp (1982), focused on fully-connected symmetric networks. These networks had trunk groups connecting every switch with each other, and all trunk groups had the same number of trunks and traffic patterns. The trunk reservation parameters used were uniform across the network, either as a fixed percentage of the size of the trunk or a fixed integer value. Determining the trunk reservation parameters for networks with these configurations is a relatively simple process since they only had to solve for one variable.

These assumptions of uniformity and symmetry are unsuitable for TELUS' networks as TELUS' Edmonton network is not uniform. Furthermore, the data obtained for this study showed that the demand patterns on the network differs depending on the time-of-day and service areas. Therefore, we decided that trunk group specific reservation parameters are required.

Chapter 2: Literature Review and Preliminaries

2.1 Introduction

In Section 2.2, we provide a literature review on past research using trunk reservation. We also provide a brief survey on three different combinatorial optimization heuristics in Section 2.3 that were considered as solution methodologies for the application of the trunk reservation model to TELUS' Edmonton network.

2.2 Literature Review on Trunk Reservation

Trunk reservation is an appealing and simple control mechanism, either in networks or queueing systems. It is also intuitively sound as shown by Kelly (1990) through Markov decision theory analysis of a single trunk group. The trunk group is offered two classes of traffic: priority (first-routed) and non-priority (non-first-routed). The transitions of the Markov decision process (MDP) are dependent on the arrival rates of the two types of traffic. A reward of r_1 is realized if a priority call is connected on the trunk group, while reward r_2 is obtained if a non-priority call is connected, with $r_1 > r_2 > 0$. The states of the MDP are the number of trunks occupied in each trunk group, and the action is to either connect or not connect an incoming call. Finally, the objective is to maximize the long-run average expected reward per unit time. The resulting optimal policy of the MDP is to accept priority calls if the trunk group is not full and to accept non-priority calls only if the number of trunks available in the trunk group is above an optimal value (Lippman 1975, Miller 1969). This optimal policy is essentially a trunk reservation policy.

The cost-efficient method of connecting telephone calls is to route them directly, which only requires one link per call. This allows for better utilization of the trunks in the network. A call routed on a 2-link tandem path occupies trunks that two other calls could use, as shown in Section 1.5. In practice, however, it is undesirable to engineer a network where all calls are routed directly because of variation in the traffic loads due to service areas (business versus residential) and seasonal variations (Ash, Cardwell and

Murray 1981). An example of seasonal variation is switches servicing universities, where demand is higher during school term versus the summer months.

Dynamic routing takes advantage of variations in traffic load effectively by adjusting routing patterns in accordance with varying and uncertain offered traffic. The constantly changing routing rules make better use of spare capacities in the network and provide extra flexibility and robustness to respond to failures or overloads (Gibbons and Kelly 1990). Dynamic routing, however, is not possible in TELUS' current networks because the switches in their networks are not homogeneous and hence, some switches do not possess the capability to support it. Instead, a variation on dynamic routing called alternate time-of-day routing is used, where the alternate routing tables change depending on the time of day. Dynamic routing systems also incorporate a form of trunk reservation in their routing-decision processes. Dynamic routing automatically regenerates new alternate paths for an origin-destination pair when the current multi-link paths for that origin-destination pair are approaching operational capacities.

Trunk reservation is beneficial especially when the network is under overload conditions because it prevents runaway congestion (Simons 1997). It reduces the overall loss probabilities or blocking levels and provides stability in the network during the busy periods by reducing the knock-off effects of tandem paths. Gibbons and Kelly (1990) found that using a small non-zero value for the trunk reservation parameter (1 to 5 trunks) significantly improves the loss probability by limiting multi-link calls. Increasing the trunk reservation parameter any higher provides relatively little additional benefits to the system.

Akinpelu (1984) provided some empirical results on the carried loads for three networks of different sizes (25, 30 and 140 switches) operating with and without trunk reservation. In the systems with trunk reservation, 5% of the trunks in each trunk group was reserved for first-routed traffic with a minimum of one reserved trunk. In all of the three test networks, the trunk reservation control increased the carried load on the network and decreased the average number of trunks used per call.

2.3 Combinatorial Optimization Heuristics

There are many heuristics to solve difficult combinatorial optimization problems. Three widely known and often used heuristics are tabu search, simulated annealing and genetic algorithms.

2.3.1 Tabu Search

The tabu search algorithm searches for the optimal solution using greedy heuristics, and incorporates a way for the algorithm to get out of local optima when it reaches one. Tabu search begins as a greedy local search algorithm. When it reaches an optimal point, be it local or global, it does not stop and report the solution. Instead, it generates other possible moves or neighbours and chooses the move that degrades the objective function the least in the hopes that the algorithm jumps into another area where it can find the global optimum.

However, after reaching a local optimum, the next least costly move may place the algorithm at a point where a local search will return it to the same local optimum point. To prevent this potential cycling within the same local optimum region, a tabu list is used. For every move from one neighbour to another, the move (and its reverse move) is placed in the tabu list and is kept tabu (restricted or cannot be used) for a set duration (tenure). Tenures in the tabu list are typically stated in terms of the number of iterations. There is one exception for shortening the tenure of a move in the tabu list. If the algorithm can achieve a solution that is better than any other solution obtained so far by using a move currently in the tabu list, this move becomes eligible and the algorithm will use it. This exception is called an aspiration criterion.

Wolsey (1998) provides the following basic version of the tabu search algorithm:

1. Initialize an empty tabu list.
2. Get an initial solution S .

3. While the stopping criterion is not satisfied:
 - 3.1 Choose a subset $Q'(S) \subseteq Q(S)$ of non-tabu solutions.
 - 3.2 Let $S' = \arg \min\{f(T): T \in Q'(S)\}$.
 - 3.3 Replace S by S' and update the tabu list.
4. On termination, the best solution found is the heuristic solution.

Glover (1989, 1990) provides a broader discussion on tabu search.

2.3.2 Simulated Annealing

Metropolis, Rosenbluth, Rosenbluth, Teller and Teller (1953) were the first to publish the concept behind simulated annealing. It is based on a process called annealing, which is the cooling of a material in a heat bath. When a solid material is heated past its melting point and then cooled back, the quality of the cooled solid depends on the rate of cooling. Thirty years later, Kirkpatrick, Gelatt and Vecchi (1983) and Cerny (1985) applied this methodology to several combinatorial optimization problems, including the travelling salesman problem and the optimal design of computers. In an optimization problem, the objective function and the variables of the problem correspond to the energy level and the molecules of the solid material respectively. The aim for the algorithm is to converge to the globally optimum solution when the process reaches the "cooled state".

Simulated annealing, like tabu search, accepts moves that improve the objective function and has a means of moving away from locally optimal regions. At each iteration of the simulated annealing algorithm, a neighbour is randomly generated. If the proposed move improves the objective function, the move is accepted. If the proposed move worsens the value of the objective function, the move is accepted based on a certain probability. The probability of accepting a worse solution is proportional to the degradation of the objective function. Slightly worse solutions will have a higher probability of being accepted because it does not drastically change the value of the objective function. Conversely, proposed moves that deviate significantly from the current value of the objective function are rarely accepted. If the number of iterations is large, the algorithm can move away from all local optima. On the other hand, for the

algorithm to eventually converge to a “good” local optimum value, the probability of accepting worse solutions should decrease as the number of iterations increase (Wolsey 1998).

Wolsey (1998) provides the following outline for the simulated annealing algorithm:

1. Get an initial solution S .
2. Get an initial temperature T and a reduction factor r with $0 < r < 1$.
3. While not yet frozen, do the following:
 - 3.1 Perform the following loop L times:
 - 3.1.1 Pick a random neighbour S' of S .
 - 3.1.2 Let $\Delta = f(S') - f(S)$.
 - 3.1.3 If $\Delta \leq 0$, set $S = S'$.
 - 3.1.4 If $\Delta > 0$, set $S = S'$ with probability $e^{-\Delta/T}$.
 - 3.2 Set $T \leftarrow rT$.
4. Return the best solution found.

The acceptance conditions in steps 3.1.3 and 3.1.4 are known as the Metropolis criterion, evolving around the function Δ . The variable r , which decreases the temperature T , reduces the probability of accepting worse solutions as the number of iterations increases. Several books and papers have been published on the theory and application of simulated annealing including van Laarhoven and Aarts (1987), Aarts and Korst (1989) and Johnson, Aragon, McGeoch and Schevon (1989, 1990).

2.3.3 Genetic Algorithm

The genetic algorithm is based on the evolutionary process of biological organisms in nature (Beasley and Chu 1996). Starting from a population of individuals, fitter individuals have a better chance of reproducing and surviving. Therefore, the genes from fitter individuals will be transferred to offsprings in future generations. In time, the population will carry good genes.

Transferring this idea to combinatorial optimization problems, the individuals in the population is the set of feasible solutions. The fitness levels of individuals correspond to their values in the objective function. Mating frequencies for an individual is proportional to its objective function value relative to the sum of all the individuals in the population. During mating, genes are exchanged between a pair of individuals to produce two new offsprings. After mating, some randomness may be introduced into the process by mutating the genes of the offsprings. With each successive generation, the size of the population grows very quickly. To maintain a constant size for the population, the parents or those individuals with fitness levels in the bottom half of the population may be eliminated after the mating stage.

Beasley and Chu (1996) provides the following outline for the genetic algorithm:

1. Generate an initial population.
2. Evaluate fitness of individuals in the population.
3. Repeat the following until a satisfactory solution has been found:
 - 3.1 Select parents from the population.
 - 3.2 Recombine (mate) parents to produce children.
 - 3.3 Evaluate fitness of the children.
 - 3.4 Replace some or all of the population by the children.

Further discussions on genetic algorithms are available in Goldberg (1989) and Reeves (1993).

Chapter 3: Modeling the Trunk Reservation Network

3.1 Introduction

Fixed-point approximation of a birth-death process is used to model the trunk reservation network to calculate the overflow levels for different combinations of trunk reservation parameters. Section 3.2 describes how we modeled the trunk reservation problem for a single trunk group using a birth-death process. The associated stationary probability distributions for the birth-death process are provided in Section 3.3. Section 3.4 describes the data provided for this study from TELUS' Edmonton network and the assumptions made on the data in our application of this study to the Edmonton network. Data for non-first-routed traffic, however, is not available in the Edmonton database and we propose a heuristic in Section 3.5 to determine the arrival rates for non-first-routed traffic. Section 3.6 discusses an assumption used in our derivation of non-first-routed traffic in Section 3.5.

3.2 A Birth and Death Process

Figure 6 shows the transition diagram of a birth-death process to model a single trunk group. The states of a birth-death process of a trunk group are the number of trunks occupied or busy at any point in time. State 0 represents the state where none of the trunks in the trunk group is occupied, state 1 represents the state where only one trunk is occupied, state 2 as two trunks occupied, and so on. The size of the trunk group is c trunks, with r trunks reserved for first-routed traffic. (First-routed traffic are calls that connect on the first path of a route, usually the direct path, between two switches, while non-first-routed traffic are calls that connect through tandem or multi-link paths.)

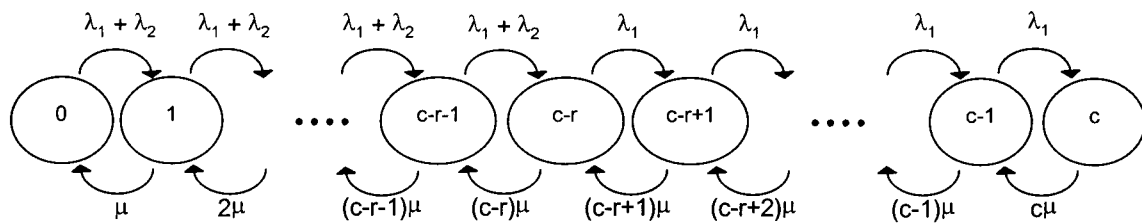


Figure 6: Transition diagram of a birth-death process to model a single trunk group

The transition from one state to another depends only on the current state and the subsequent action in the trunk group. Arrivals into the trunk group, identified by the arrival or birth rates λ_1 and $\lambda_1 + \lambda_2$ in Figure 6, increase the number of occupied trunks. The arrival rates λ_1 and $\lambda_1 + \lambda_2$ are trunk specific, that is, the values of λ_1 and $\lambda_1 + \lambda_2$ depends on the trunk group being analyzed. When a call is completed or terminated in the trunk group, the number of occupied trunks decreases. The rate at which the number of occupied trunks decreases is shown as μ in Figure 6 and is called the service or death rate. Again, the value of μ is trunk group specific.

In Figure 6, first-routed and non-first-routed traffic are identified by the arrival rates λ_1 and λ_2 respectively. The total arrival rate onto the trunk group at all times is actually $\lambda_1 + \lambda_2$. The differentiation of arrival rates is necessary because of the routing properties of trunk reservation. If there are unreserved trunks available to connect calls through this trunk group, the transition rate of $\lambda_1 + \lambda_2$ is used. This occurs when the trunk group is in any of states 0 to $c-r-1$ inclusive. When all unreserved trunks are occupied (states $c-r$ to $c-1$), the number of occupied trunks in the trunk group increases only when first-routed traffic enters at the rate of λ_1 .

At this point, we need to revisit the differences between the terms blocking and overflow in a network. Generally, a call that cannot form a circuit or be connected through a path is not blocked but is overflowed onto the next path in the route. When a call overflows from the final path in its routes, the call is then blocked. Since the trunk reservation problem is modeled on a trunk group basis, the corresponding analysis should also be performed on a trunk group basis.

In a trunk group, a call overflows under two conditions. Firstly, if a non-first-routed call arrives to a trunk group when the trunk group is in any of the states from $c-r$ to $c-1$, the call will overflow because all the available trunks are reserved for first-routed traffic. The other situation is when the trunk group is in state c (all the trunks in the trunk group are occupied). Regardless if the call is first-routed traffic or not, it will overflow because there are no more available trunks.

3.3 Stationary Probability Distributions

The next step is to determine the proportion of time a trunk group spends in each of its states given the arrival rates, service rate, and the number of reserved and unreserved trunks. If a trunk group is observed for an extended period, the proportion of time it spends in a state converges to a certain percentage of the total observation time. The collection of percentages is called the stationary probability distributions (or long-run probabilities or limiting distributions) of a trunk group being in different states.

The stationary probability distribution for a birth-death process is obtained by formulating the balance equations for each state. (See Ross 1996.) Balance equations equate the rates at which the process leaves and enters a state. The balance equations for the states of a trunk group are:

$$\begin{aligned}\mu\pi_1 &= (\lambda_1 + \lambda_2)\pi_0 \\ (\lambda_1 + \lambda_2)\pi_{(i-1)} + (i+1)\mu\pi_{(i+1)} &= (\lambda_1 + \lambda_2)\pi_i + i\mu\pi_i, & i = 1, 2, \dots, c-r-1 \\ (\lambda_1 + \lambda_2)\pi_{(i-1)} + (i+1)\mu\pi_{(i+1)} &= \lambda_1\pi_i + i\mu\pi_i, & i = c-r \\ \lambda_1\pi_{(i-1)} + (i+1)\mu\pi_{(i+1)} &= \lambda_1\pi_i + i\mu\pi_i, & i = c-r+1, c-r+2, \dots, c-1\end{aligned}$$

where $\sum_i \pi_i = 1$.

Solving the system of equations above in terms of π_0 gives:

$$\pi_i = \begin{cases} \pi_0 \left(\frac{\lambda_1 + \lambda_2}{\mu} \right)^i \frac{1}{i!} & \text{if } 0 \leq i \leq c-r \\ \pi_0 (\lambda_1 + \lambda_2)^{c-r} \frac{\lambda^{i-c+r}}{\mu^i \cdot i!} & \text{if } c-r \leq i \leq c \end{cases} \quad (1)$$

where

$$\pi_0 = \left[\sum_{i=0}^{c-r} \left(\frac{\lambda_1 + \lambda_2}{\mu} \right)^i \frac{1}{i!} + \sum_{i=c-r+1}^c (\lambda_1 + \lambda_2)^{c-r} \frac{\lambda^{i-c+r}}{\mu^i \cdot i!} \right]^{-1}, \quad (2)$$

π_i is the stationary probability of the trunk group being in state i ,
 λ_1 is the arrival rate of first-routed traffic into the trunk group,
 λ_2 is the arrival rate of non-first-routed traffic into the trunk group,
 μ is the mean service rate of a call in the trunk group,
 c is the total number of trunks in the trunk group, and
 r is the number of reserved trunks in the trunk group.

The overflow probability for first-routed traffic is therefore

$$\pi_c \tag{3}$$

and that for non-first-routed traffic is

$$\sum_{i=c-r}^c \pi_i . \tag{4}$$

The total amount of overflow traffic on a trunk group is calculated as

$$\lambda_1 \pi_c + \lambda_2 \sum_{i=c-r}^c \pi_i \tag{5}$$

where π_i is dependent on the parameters of the trunk group: the number of trunks and reserved trunks, and the arrival and service rates of calls.

We now have a model to evaluate the effects of different values of r (the number of reserved trunks) to a single trunk group. We then state the following decomposition assumption: analyzing the network on a trunk group by trunk group basis is equivalent to analyzing the network as a whole. With this assumption, we can model the entire trunk reservation network using fixed-point approximations of a birth-death process (Akinpelu 1984, Gibbons and Kelly 1990, Chen 2001).

The objective for this study, therefore, is to find the values for r (the number of reserved trunks) for each trunk group to allow the maximum number of calls to be connected in the network. By minimizing the total amount of overflow from each trunk group, we are invariably routing as many calls as possible through their direct paths. However, the final value of r will also incorporate the opportunities to route calls through underutilized portions of the network. (See Ash et al. 1981 and Smith 2000.)

Therefore, we propose to minimize the total overflow level across the network as our objective, where the total overflow level is the sum of the overflow in every trunk group in the entire network. An alternative and much preferred objective function for this trunk reservation problem is to minimize the amount of blocked calls in the network. However, as we have chosen to model the trunk reservation problem on a trunk group basis, the results of minimizing blocked calls may be inaccurate.

3.4 Modeling the Arrival and Service Rates

The arrival and service rates of telephone calls for the Edmonton data set can be modeled on an origin-destination basis using Poisson and exponential distributions respectively. (See Baird 2001.) Baird also found that a mixture of two exponential distributions provide for a better fit for the service distributions. Our modeling of the trunk reservation problem requires only a single value for μ , the mean service rate. Using this single value may not accurately capture the benefits of the mixture distribution and this could cause our results to be slightly inaccurate since we will be approximating overflow using an approximate input parameter.

The trunk reservation model discussed in Section 3.2 requires, as inputs, a service rate μ and the arrival rates λ_1 and λ_2 on a trunk group basis. The trunk group arrival rate λ_1 is equal to the Poisson arrival rates derived by Baird. However, we only have μ on an origin-destination basis and the arrival rate λ_2 is not readily available.

Since TELUS' Edmonton network is a direct routing network, the fitted theoretical service rates on an origin-destination basis should therefore be the service rates on a trunk-

group basis. However, the trunk reservation network in our study uses alternate routing. The percentage of calls connecting on multi-link paths, though, is small – less than 2%. (See Figure 17 for the percentage of multi-link calls in the no reservation network at the load scale factor of 1.). Hence, we assume that alternate routing will not significantly modify the distributions of the average service rates in the trunk groups and we use the respective origin-destination mean service rates μ for the trunk group service rates in our analysis.

3.5 Calculating the Non-First-Routed Traffic Arrival Rates

Determining the non-first-routed traffic λ_2 onto each trunk group is just as difficult. However, queueing theory allows for better approximations in the calculation of the arrival rates. Before proceeding any further, some notation is required. Define λ_1^{ij} and λ_2^{ij} to be the first-routed and non-first-routed traffic arrival rates onto trunk group ij respectively. Let R_{ij} to be the set of paths connecting originating switch i and destination switch j :

$$R_{ij} = \{ p_1^{ij}, p_2^{ij}, \dots, p_{N_{ij}}^{ij} \} \quad (6)$$

where p_k^{ij} is the path connecting origin-destination switches ij and has route index k (see Section 1.2), and N_{ij} is the number of paths in the route ij . In Figure 7, $R_{AD} = (A \rightarrow D, A \rightarrow B \rightarrow D, A \rightarrow C \rightarrow D)$, $R_{CD} = (C \rightarrow D)$ and $R_{ED} = (E \rightarrow D, E \rightarrow C \rightarrow D)$ with the number of paths N for each route being three, one and two respectively. Trunk group CD forms p_1^{CD} in R_{CD} , the second leg of p_3^{AD} in R_{AD} , and the second leg of p_2^{ED} in R_{ED} .

Next, define $\lambda_{ij}(k)$ to be the arrival rate contributed to trunk group ij by all k -th paths that contains ij , regardless of the origin or destination of the call. First-routed traffic through a trunk group ij , λ_1^{ij} , is determined by considering all first paths for calls from every origin-destination pair using that trunk group. In the Edmonton network, the first path for each origin-destination pair is the direct path. Therefore, the first-routed arrival

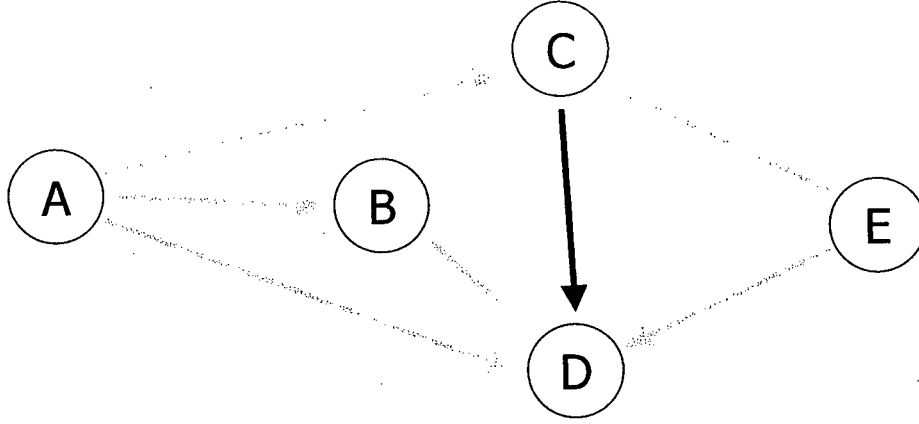


Figure 7: An example to illustrate the calculation of the total arrival rate onto trunk group CD

rate λ_1^{ij} for the trunk group ij is simply the sum of the arrival rates from switches i to j and from switches j to i , that is:

$$\lambda_1^{ij} = \lambda_{ij}^{(1)} + \lambda_{ji}^{(1)}. \quad (7)$$

The total non-first-routed traffic arrival rate for a trunk group ij , λ_2^{ij} , is the sum of the arrival rates from the paths in which trunk group ij forms a segment of the path. If the routes for all origin-destination switches ij contain four paths, we define λ_2^{ij} as:

$$\lambda_2^{ij} = \lambda_{ij}^{(2)} + \lambda_{ij}^{(3)} + \lambda_{ij}^{(4)}, \quad (8)$$

where $\lambda_{ij}^{(k)}$ the arrival rate contributed to trunk group ij by all k -th paths in which ij is a segment. Equation (8) and all subsequent formulation discussions assume that each route contains at most four paths but the approach outlined here is not constrained to only four paths. It is applicable regardless of the maximum number of paths in a route.

Since the value of λ_2^{ij} is unknown, a heuristic was constructed to iteratively find this value (Chen 2001). The value of λ_2^{ij} is likely a positive non-zero value since networks are typically engineered for a P.01 service rate (no more than one out of 100 calls are blocked). At a P.01 service rate in an alternate routing network, some calls will overflow from their first paths onto their alternate paths. These overflowed calls form the non-

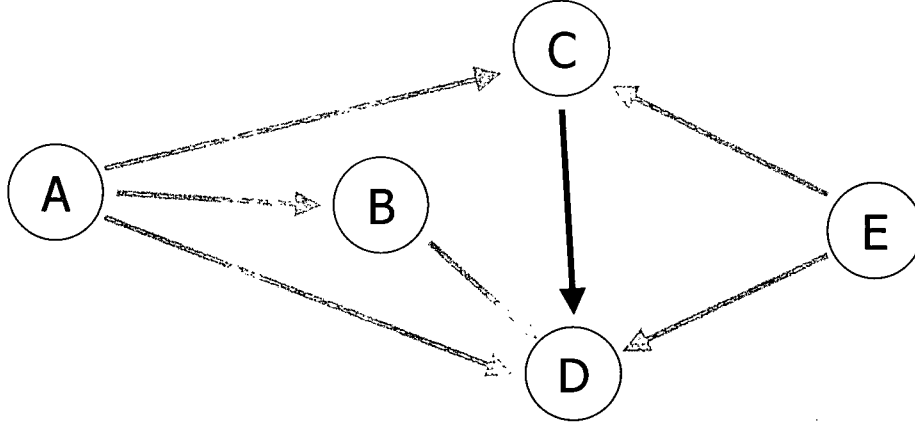


Figure 7: An example to illustrate the calculation of the total arrival rate onto trunk group CD

rate λ_1^{ij} for the trunk group ij is simply the sum of the arrival rates from switches i to j and from switches j to i , that is:

$$\lambda_1^{ij} = \lambda_{ij}^{(1)} + \lambda_{ji}^{(1)}. \quad (7)$$

The total non-first-routed traffic arrival rate for a trunk group ij , λ_2^{ij} , is the sum of the arrival rates from the paths in which trunk group ij forms a segment of the path. If the routes for all origin-destination switches ij contain four paths, we define λ_2^{ij} as:

$$\lambda_2^{ij} = \lambda_{ij}^{(2)} + \lambda_{ij}^{(3)} + \lambda_{ij}^{(4)}, \quad (8)$$

where $\lambda_{ij}^{(k)}$ the arrival rate contributed to trunk group ij by all k -th paths in which ij is a segment. Equation (8) and all subsequent formulation discussions assume that each route contains at most four paths but the approach outlined here is not constrained to only four paths. It is applicable regardless of the maximum number of paths in a route.

Since the value of λ_2^{ij} is unknown, a heuristic was constructed to iteratively find this value (Chen 2001). The value of λ_2^{ij} is likely a positive non-zero value since networks are typically engineered for a P.01 service rate (no more than one out of 100 calls are blocked). At a P.01 service rate in an alternate routing network, some calls will overflow from their first paths onto their alternate paths. These overflowed calls form the non-

first-routed traffic, λ_2^{ij} . In our heuristic, we initialize the value of λ_2^{ij} to its lower bound value of zero for all ij . λ_2^{ij} is updated through a series of iterative steps until it eventually converges, which we assume this value to be a good approximation of λ_2^{ij} . Before the application of this heuristic, formulas to calculate the components of λ_2^{ij} in equation (8) are required.

The arrival rate contributed to any trunk group ij by second paths is the sum of the overflow of call arrivals on the first paths where trunk group ij is a link on the second paths of these calls:

$$\lambda_{ij}^{(2)} = \sum_{k, l: (i, j) \in p_{kl}^{(2)}} \lambda_1^{kl} \cdot \pi_{c_{kl}}^{kl}(\lambda_1^{kl}, \lambda_2^{kl}) \quad (9)$$

where $\pi_{c_{kl}}^{kl}(\lambda_1^{kl}, \lambda_2^{kl})$ denotes the long-run probability that trunk group kl is in state c_{kl} , which is calculated from equation (1).

The formula to compute $\lambda_{ij}^{(3)}$ – the total arrival rate for trunk group ij contributed by all third paths where ij is a segment of the third path in route kl – is as follows:

$$\lambda_{ij}^{(3)} = \sum_{k, l: (i, j) \in p_{kl}^{(3)}} \lambda_1^{kl} \cdot \pi_{c_{kl}}^{kl}(\lambda_1^{kl}, \lambda_2^{kl}) \cdot \left[1 - \prod_{(x, y) \in p_{kl}^{(2)}} \left(1 - \sum_{n=c_{xy}-r_{xy}}^{c_{xy}} \pi_n^{xy}(\lambda_1^{xy}, \lambda_2^{xy}) \right) \right] \quad (10)$$

where the first term in the first summation is the first-routed traffic arrival rate onto trunk group kl , the second term calculates the overflow from the first path onto the second path, and the third term (everything in the square brackets) calculates the overflow from the second path onto the third path. Overflow of calls from the second path onto the third path can occur in any of the legs of the path. For example, if the second path consists of two trunk groups, one of the two trunk groups may be fully occupied. Since a circuit cannot be formed through the fully occupied trunk group, and the call overflows. The formula in the square bracket first calculates the probability of a call connecting on the second path (product sign) and then, the probability of a call

overflowing from the path. Note that the summation within the square brackets is identical to equation (4).

Similarly, the formula to compute $\lambda_{ij}^{(4)}$ is:

$$\lambda_{ij}^{(4)} = \sum_{k,l: (i,j) \in p_{kl}^{(3)}} \lambda_1^{kl} \cdot \pi_{c_{kl}}^{kl}(\lambda_1^{kl}, \lambda_2^{kl}) \cdot \left[1 - \prod_{(x,y) \in p_{kl}^{(2)}} \left(1 - \sum_{n=c_{xy}-r_{xy}}^{c_{xy}} \pi_n^{xy}(\lambda_1^{xy}, \lambda_2^{xy}) \right) \right] \cdot \left[1 - \prod_{(x,y) \in p_{kl}^{(3)}} \left(1 - \sum_{n=c_{xy}-r_{xy}}^{c_{xy}} \pi_n^{xy}(\lambda_1^{xy}, \lambda_2^{xy}) \right) \right] \quad (11)$$

This formulation can be expanded further to calculate any value of $\lambda_{ij}^{(k)}$ for $k > 4$.

We now formally outline the heuristic to calculate λ_2 :

1. Set $\lambda_2^{ij} = 0$ for all ij and $\varepsilon = 0.00001$.
2. Repeat
 - 2.1 Compute $\lambda_{ij}^{(2)}$, $\lambda_{ij}^{(3)}$ and $\lambda_{ij}^{(4)}$ for all ij using equations (9), (10) and (11).
 - 2.2 Calculate $(\lambda_2^{ij})'$ for all ij using equation (8) from step 2.1.
 - 2.3 For each ij , if the value of $(\lambda_2^{ij})'$ is greater than the current value, replace λ_2^{ij} with $(\lambda_2^{ij})'$.
3. Until $|\lambda_2^{ij} - (\lambda_2^{ij})'| < \varepsilon$ for all ij .

In a preliminary test, the heuristic converged after eight iterations, taking two seconds running on a Pentium 4 1.4Ghz computer. Although a better initial value for λ_2^{ij} can be found to potentially reduce the runtime, initializing λ_2^{ij} to zero is sufficient for the purpose of this study.

3.6 Independence of Paths in a Route

In the formula to calculate $\lambda_{ij}^{(k)}$, there is an inherent assumption that none of the paths in route ij share a similar trunk group, that is, all the paths in a route are independent of each other. Consider the case where trunk group ij is the first leg for both the

second and third paths of an origin-destination pair. If a call overflows on the first leg of the second path, the same call will again overflow on the first leg of the third path. These two events are therefore not mutually independent and the formulation for $\lambda_{ij}^{(k)}$ would have to be modified.

However, in a study by Ash et al. (1981), they simulated a dynamic routing network allowing for multi-link paths – paths consisting of more than two trunk groups. They found that 98% of the paths used to route calls were either one-link or two-link paths. When restricting the paths generated by their dynamic routing network to either one or two-link paths, the results obtained were similar to their initial study with multi-link paths. Restricting the number of links in a path to at most two links provides three benefits:

- (i) the generation of alternate paths for routes becomes a simpler task;
- (ii) reduces delay in relaying of tandem calls in the switches (a possible bottleneck);
- (iii) multi-link paths with at least three links use more capacity and increase the probability of overflow; and,
- (iv) more importantly, none of the paths in a route will share a common link, which validates our independence assumption.

Chapter 4: Problem Formulation and Solution Methodology

4.1 Introduction

The objective of this study is to find trunk specific reservation parameters for a non-symmetric network with non-uniform demand, for example, TELUS' Edmonton network. This is a typical optimization problem where the objective is to minimize the total overflow values across the network, with the variables being the number of reserved trunks in each trunk group. Section 4.2 discusses the choice of using heuristics as the solution methodology over mathematical optimization and simulation. Section 4.3 outlines the specific details to solving this trunk reservation problem using the simulated annealing algorithm.

4.2 Heuristics, Mathematical Optimization and Simulation

Heuristics are techniques that find near-optimal solutions at a reasonable cost (Reeves 1993). Compared to mathematical programming, they offer excellent trade-offs between solution quality, computational time and ease of implementation. Moreover, heuristics provide the modeler with the flexibility to incorporate more complicated and, most importantly, more realistic objective functions and/or constraints into the model (Pirlot 1996). Flexibility is important when solving real-world problems because the problems are often not as cut-and-dried for mathematical programming models. Furthermore, the optimal solution obtained from the assumption-filled mathematical model is not necessarily the best solution to the underlying real-world problem. The question then becomes: "is an exact solution of an approximate model preferred over an approximate solution of an 'exact' model?" (Reeves 1993)

For the trunk reservation problem, an approximate solution to an exact model is more desirable because of the complexity of the problem. Moreover, the problem is a non-linear discrete optimization problem – with integrality constraints on the variables (trunk reservation parameters) and non-linearity in the calculation of π_i . These types of problems are difficult to solve and optimality is often not guaranteed. Therefore, heuristics is preferred as the solution methodology over mathematical programming

approaches. A network simulator (a deliverable provided to TELUS from another project completed by the COE) will be used to analyze the quality of the trunk reservation solution obtained from the heuristic.

The network simulator could have been used to generate the trunk reservation parameters but this method is cumbersome. The run-time to simulate one scenario of the Edmonton network takes, on average, 10 minutes. In the fully-connected Edmonton network, there are 66 trunk groups connecting the 12 switches. To check if there are any better configurations (providing low overflow levels) at the end of a simulation run, we would have to enumerate all possible neighbours to this current solution. The neighbours here are generated by increasing or decreasing the number of reserved trunks in a trunk group in the current solution. The simulation is executed again to observe the effect of the change on the network. This is an intractable task since at any starting network configuration, there are 2^{66} or 7.4×10^{19} possible scenarios to analyze (assuming the number of reserved trunks in every trunk group in the current solution is positive). At an average rate of 10 minutes per scenario, the total run time to evaluate all the 2^{66} possible moves from the current solution is approximately 14 trillion centuries! Furthermore, we would also need to account for statistical errors in the results from the simulations.

Many algorithms can be used to solve these types of difficult optimization problems. The simulated annealing algorithm, however, seems to be the best-suited algorithm for the trunk reservation problem. Unlike genetic algorithms, which requires an initial population of feasible solutions, simulated annealing only requires a feasible solution to start the algorithm. After modeling the trunk reservation problem, a feasible solution already exists to the problem, that is, no reserved trunks in any of the trunk groups. Therefore, there is no need for an additional process to find an initial starting feasible solution.

Compared to tabu search, simulated annealing is easier to implement. Furthermore, the memory requirements to run the simulated annealing algorithm is minimal compared to tabu search. In the application of tabu search to this trunk reservation problem, the size

of the tabu list is proportional to the number of variables in the problem. For a medium-sized network (about 30 switches), a significant amount of computer memory is required to maintain the tabu list. There is also the possibility that a lot of computer processing time could be wasted searching through the tabu list of restricted moves. In addition, tabu search requires generation of a set of proposed moves from which the algorithm chooses its next neighbour. Simulated annealing only generates one neighbour for evaluation at each iteration of the algorithm.

4.3 Simulated Annealing: Problem Specific Details

4.3.1 Simulated Annealing Algorithm

The simulated annealing algorithm was first published by Metropolis et al. (1953) based on a process called annealing, which is the cooling of a material in a heat bath. Wolsey (1998) provides the following outline for the simulated annealing algorithm:

1. Get an initial solution S .
2. Get an initial temperature T and a reduction factor r with $0 < r < 1$.
3. While not yet frozen, do the following:
 - 3.1 Perform the following loop L times:
 - 3.1.1 Pick a random neighbour S' of S .
 - 3.1.2 Let $\Delta = f(S') - f(S)$.
 - 3.1.3 If $\Delta \leq 0$, set $S = S'$.
 - 3.1.4 If $\Delta > 0$, set $S = S'$ with probability $e^{-\Delta/T}$.
 - 3.2 Set $T \leftarrow rT$.
4. Return the best solution found.

We now define the components for the simulated annealing algorithm for the trunk reservation problem.

4.3.2 Objective Function, $f(S)$

In the trunk reservation problem, the objective is to find the optimal configuration of reserved and unreserved trunks that will minimize total overflow in the network. In

terms of the simulated annealing algorithm, optimality is achieved when the total overflow level remains constant over several iterations of the algorithm; that is, it has reached the frozen state.

4.3.3 Neighbourhood

The neighbourhood for the problem is defined to be the network configuration from increasing or decreasing by one the number of reserved trunks in any trunk group. We chose to change the number of reserved trunks by one because past studies on trunk reservation have suggested only a small percentage (at most 5%) of the trunk group size should be reserved for first-routed traffic only. Since over 90% of the trunk groups in the Edmonton network have capacities of less than 900 trunks, this translates to at most five reserved trunks at the suggested 5% reserved trunks maximum. To generate a neighbour, a trunk group is randomly selected. Its number of reserved trunks is either increased or reduced by one, while satisfying the constraints that the number of reserved trunks is non-negative and is at most the total number of trunks in the trunk group.

4.3.4 Initial Temperature, T

The initial value of the temperature should be one where virtually all moves, whether they improve or deteriorate the solution, are accepted at the outset of the algorithm. One method is to find the initial temperature T where $100(\chi)\%$ of the deteriorating moves are accepted. The initial temperature is calculated from

$$\chi = \exp\left(\frac{\hat{t}}{T}\right) \quad (12)$$

where $\hat{t}$ is the average increase in cost and χ is the acceptance probability (van Laarhoven and Aarts 1987).

To calculate the initial temperature, we begin with a trunk reservation model without any trunks reserved for first-routed traffic. Neighbourhoods are generated until one

whose total overflow level is greater than that of the no-reservation model is obtained. We repeat this process of generating worse solutions until 100 such solutions are obtained. Next, the differences in the total overflow levels between each of the 100 worse solutions and the no-reservation model are calculated, and the average of these differences is calculated. This average value becomes the variable $\hat{t}$ in equation 12. Finally we calculated the initial temperature T using a value for χ (the initial temperature acceptance probability).

In past applications of simulated annealing to other types of problems, the initial temperature acceptance probability χ was set to 80%. Since there has not been an application of simulated annealing for the trunk reservation problem, we carried out multiple annealing simulations using values of 80%, 85% and 90% for χ to determine the best initial temperature for this application.

4.3.5 Length of Loop, L

The number of iterations L in each loop can be either a fixed number, a value dependent on the time the algorithm requires to reach quasi-equilibrium at each temperature, or a minimum number of accepted moves. One drawback of using a fixed number is that an optimal or near-optimal solution may not be achieved as this schedule might be cooling the process too quickly. Therefore, the algorithm will end up in a local optimum. The second option should provide the optimal or near-optimal solution, but the definition and determination of when quasi-equilibrium is achieved is difficult. The third option is a combination of the first two. In the third option, it is intuitively clear that as the temperature decreases, the length of the loop increases rapidly. This is because the probability of accepting deteriorating moves decreases as the temperature decreases, which translates to requiring a longer loop to satisfy the requirement for a minimum number of accepted moves. In mathematical notation, $L \rightarrow \infty$ as $T \rightarrow 0$.

For our study, we chose the third option: using a minimum number of accepted moves to determine loop length. To prevent long loop lengths when using this strategy for L , several authors (Kirkpatrick et al. 1982, Johnson et al. 1991) propose imposing a cap on

the length of the loops. The cap can be a multiple of the number of variables in the problem. For this trunk reservation problem, we set the cap to 132, which is a multiple of the 66 variables (trunk groups) in the model. Therefore, the loop will terminate when it has either reached the cap or if the total number of improvements to the objective function exceeds $x\%$ of the maximum number of iterations (the cap of 132 iterations). Although the rule of thumb for x is 2% (Pirlot 1996), we ran multiple annealing simulations for x from 1% to 5% as sensitivity analysis on x .

4.3.6 Temperature Reduction Rate, r

The temperature reduction rate r should be a value close to 1. If not, the quasi-equilibrium objective function value of the current temperature will be far away from that of the next temperature, and so will require longer loop lengths in order to restore quasi-equilibrium at that subsequent temperature (Aarts and Korst 1989). Furthermore, if the algorithm uses the "minimum number of accepted moves" criteria for the loop length, the algorithm might not reach the quasi-equilibrium state before the temperature is reduced again. This might result in sub-optimal final solutions. So, there is an interaction between the reduction rate and the length of the loop, and hence, a trade-off between the two parameters (van Laarhoven and Aarts 1987).

Again, there is no set rules for the choice of r , so we ran multiple annealing simulations for values of r ranging from 0.4 to 0.9 (increments of 0.05) to find the optimal temperature reduction rate.

4.3.7 Stopping Criterion

Finally, a stopping criterion for the algorithm is required. One method is to stop when the process has reached a quasi-equilibrium state; that is, the value of the objective function is constant or within a small value ϵ of each other for a consecutive number of temperature levels (Kirkpatrick et al. 1982). Johnson et al (1991) uses a counter, which is incremented when the proportion of accepted moves to the loop length L is less than a certain percentage. (See the discussion on the minimum number of accepted moves

criteria for loop length L in section 4.3.5). The counter is reset to zero if a solution found is the best so far. When the counter reaches a pre-determined value, the annealing process is deemed frozen and the algorithm stops.

We implemented the stopping criterion used by Johnson et al. (1991), which terminates the algorithm when five consecutive loops do not meet the minimum acceptance rate for the loop length L .

Chapter 5: Results

5.1 Introduction

In Section 5.2, we provide the results from running the simulated annealing algorithm for different combination of values for the three input parameters: initial temperature acceptance rate, minimum number of accepted moves (acceptance rate) and reduction rate. All simulations of the annealing algorithm eventually converged to the same objective function value, but the values of the variables obtained were different for some trunk groups. We discuss the procedure used to obtain the final optimal reservation parameters. (A complete description of the analysis and procedure is provided in Section 5.4.) Finally, we validated the quality of the final optimal reservation parameters through a simulation and presented the results in Section 5.3.

5.2 Simulated Annealing Results

In total, we simulated 165 annealing processes, corresponding to the three initial temperature acceptance rates, five acceptance rates and 11 reduction rates. In the no-reservation model, the total overflow level is approximately 11.4 (with slight deviations from this value depending on the initial temperature acceptance rate). In over 90% of the 165 annealing simulations, the final total overflow levels converged to 10.2751, while the remaining simulations ended with overflow levels slightly greater than 10.2751. These results indicate that trunk reservation reduced the total overflow level of the network by 10%.

Figure 8 shows the total overflow levels at the start of each loop for the annealing process using an initial acceptance probability of 85%, a reduction rate of 65% and an acceptance rate of 3%.

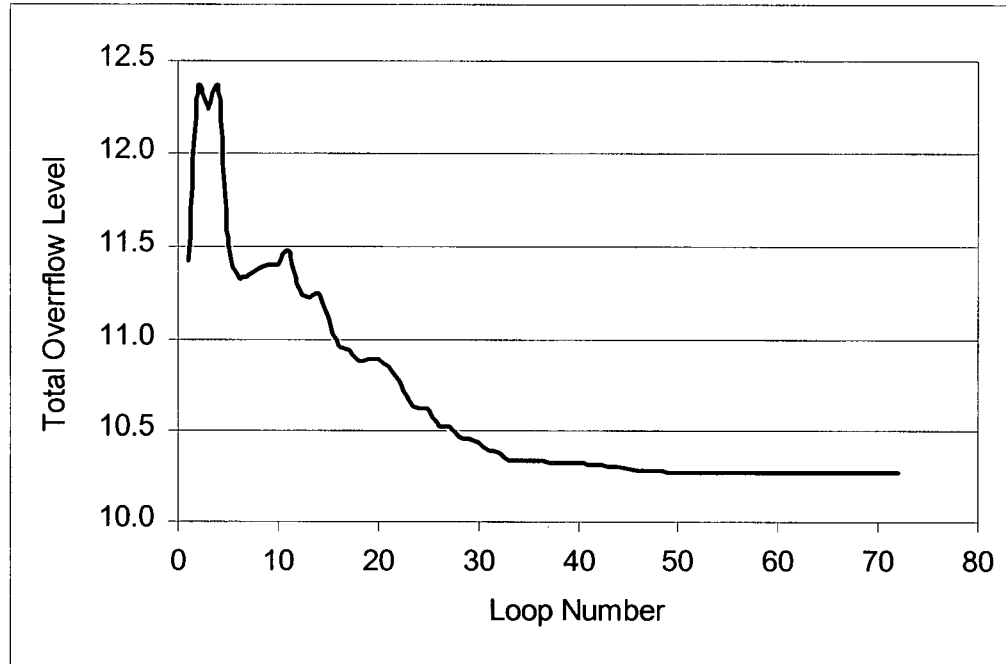


Figure 8: Evolution of the total overflow level for a simulated annealing process

On the other hand, the majority of the annealing simulations ending with the same overflow level also promote the possibility that there are multiple optimal configurations leading to the same final objective function value. Unfortunately, this possibility has manifested itself in the annealing results. We grouped the reservation parameters from the 165 annealing simulations by trunk groups, and created distributions of the number of reserved trunks for each trunk group. Of the 66 trunk groups in the network, the number of reserved trunks for 25 of the trunk groups are easily determined. Of these 25 trunk groups, the results suggested no reserved trunks for 20 of them. Figure 9 shows the distribution of the number of reserved trunks for one of the five trunk groups with non-zero suggested number of reserved trunks. From Figure 9, this trunk group connecting switches 6 and 9 should have five trunks reserved for first-routed traffic only.

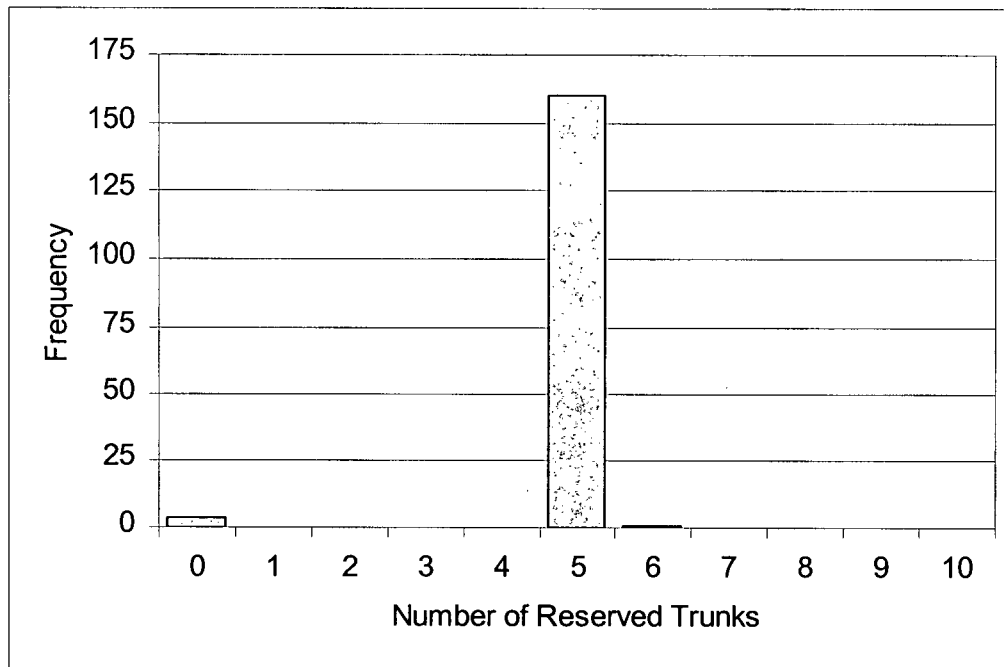


Figure 9: Distribution of the suggested number of reserved trunks for the trunk group between switches 6 and 9

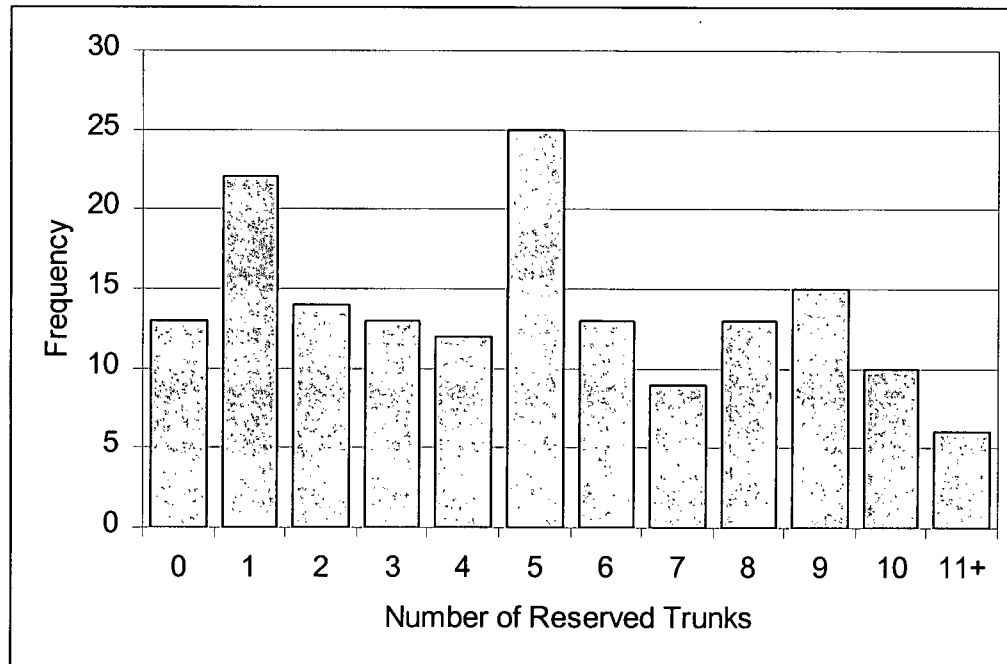


Figure 10: Distribution of the suggested number of reserved trunks for the trunk group between switches 7 and 8

It is more difficult to determine the optimal number of reserved trunks in the remaining 41 trunk groups. Figure 10 shows the distribution of the suggested number of reserved trunks for the trunk group connecting switches 7 and 8. The distribution is essentially uniformly distributed between 0 and 10.

When simulating multiple annealing processes using different values for the three annealing parameters (initial temperature acceptance rate, reduction rate and acceptance rate), some of these combinations may not be appropriate for this application. Therefore, the results from these combinations may have contributed to the uniformity in the distribution of the trunk groups. To eliminate these undesirable combinations from our results, analyses of the quality of the reservation parameters on the three annealing parameters are required.

We looked for the combination of the three annealing parameters that provide a small spread in the distribution of the number of reserved trunks in its final network. From our analysis, the results from the annealing process using an initial temperature acceptance rate of 90%, a reduction rate of 65% and an acceptance rate of 1% provided the smallest spread. (See Section 5.4 for a detailed description of the analysis.) This combination of annealing parameters converged to the same optimal total overflow level of 10.2751 and the trunk reservation parameters for each trunk group are provided in Appendix 1. We next ran this set of reservation parameters through a simulator to test the quality of this network configuration.

5.3 Simulation Results

We simulated a network containing the optimal trunk reservation parameters obtained from Section 5.2 against one without any reserved trunks, scaling the arrival rates across the network from 0.8 to 1.5 (load scale factor). A load scale factor of one represents the current demand levels for the network. We used the configurations of TELUS' Edmonton network and the fitted theoretical arrival and service rates obtained from the May study period. (See Baird 2001.) We only simulated the busy hour for the weekday period since this is the period for which the network was engineered and it is when most of the overflow occurs. For each load scale factor, we ran 30 simulations and

obtained summarized results for the total overflow level, average utilization of trunk groups and number of multi-link calls. Alternate routing was used to route the calls in both these networks. (See Kabanuk 2000 for the methodology to determine alternate routes in non-hierarchical network using linear optimization.)

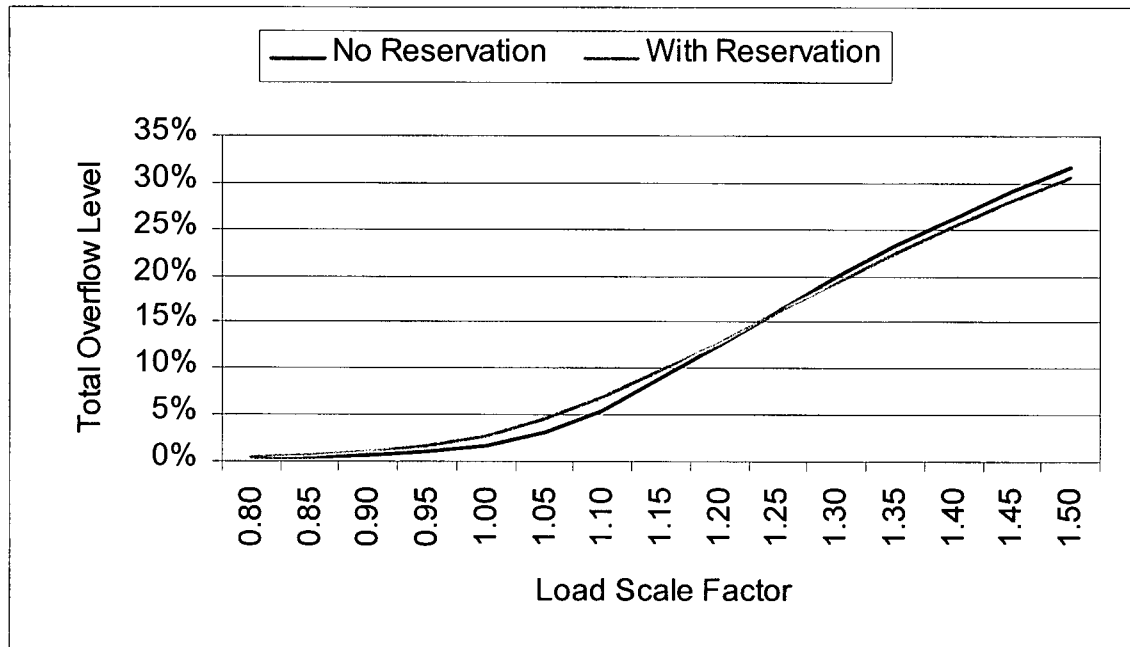


Figure 11: Total overflow levels for networks with and without trunk reservation at various load scale factors

Figure 11 shows the total overflow levels for the two networks. As the level of the demand increases in the networks, pushing them towards overload levels (where demand is greater than the capacity), the trunk reservation network begins to outperform the no-reservation network. As the network reaches overload levels, trunk reservation begins to prevent calls from connecting through multi-link paths, saving these trunk groups for direct routing. In the no-reservation network, this control is non-existent so calls are still allowed to connect on their multi-link paths, which is operationally inefficient.

Surprisingly, the no reservation network outperforms the optimal reservation network for load scale factors less than 1.25. Theoretically, this is impossible since the optimal reservation network should always outperform any other network with different

reservation parameters. Since the no-reservation network is a special case of a reservation network with all reservation parameters set to zero, the optimal reservation network should therefore outperform the no-reservation network, especially at the load scale factor of 1. In her study of trunk reservation on a symmetric network with uniform demand, Akinpelu (1984) also obtained similar results where the no-reservation network outperformed the reservation network. She states: "At overloads of less than 10%...carried load actually decreases slightly since some alternate-routed calls are prevented from accessing paths that have been engineered to carry them." The carried loads that could have been connected in the no-reservation network but could not in the optimal reservation network overflow through the network, which is reflected by a higher total overflow level in the optimal reservation network.

In Figure 11, there seems to be little difference between the total overload levels for the two networks at different load scale factors. A two-sample t-test was performed at each load scale factor. We tested the null hypothesis that the means of the total overflow level of the two networks were equal. Figure 12 shows the resulting p-values of the two-sample t-tests at each load scale factor.

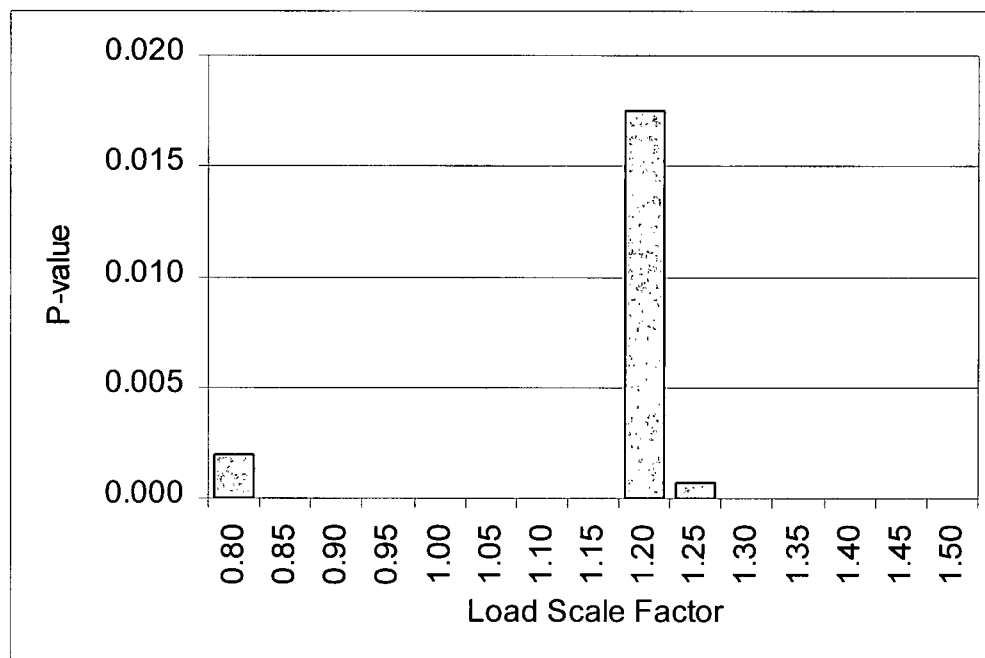


Figure 12: P-values of two-sample t-tests on the means of the total overflow levels of the no-reservation and optimal reservation networks

All the p-values are under 2%, and we can therefore reject the null hypothesis that the average total overflow level of the two networks are equal and conclude that there is indeed a difference.

However, the issue of the no-reservation network outperforming the optimal reservation network in the status quo scenario (load factor 1) still has to be addressed. There are several possible explanations for this result. Firstly, is the decomposition assumption used in the modeling of the trunk reservation problem valid? Table 2 shows the total overflow levels for the no-reservation and optimal reservation networks from the fixed-point approximation and simulation models. Fixed-point approximation reduced total overflow in the network from 11.4 to 10.3 using trunk reservation. In the simulation, however, the total overflow level increased from 1.3 to 2.4 when trunk reservation was introduced into the network. This strongly suggests that there may be a flaw in adopting the decomposition assumption when modeling the trunk reservation network using fixed-point approximations. (As an aside, the expected call blocking rate from the simulation is 1.6% (status quo scenario) compared to 6.9% in the fixed-point model.)

	Without Trunk Reservation	Optimal Trunk Reservation
Fixed-Point Approximation	11.4	10.3
Simulation	1.3	2.4

Table 2: Total overflow levels for networks with and without trunk reservation from the fixed-point approximation and simulation models

Furthermore, simulated annealing may not be the best optimization heuristic for this particular application. At each iteration of the simulated annealing algorithm, only one neighbour is generated. Choosing the best neighbour from a subset of neighbours at every iteration as in tabu search or genetic algorithm may be a better approach. Moreover, neighbours in the simulated annealing algorithm are chosen randomly, another source of variation and unpredictability that might contribute to the results of our analysis.

In the telecommunications industry, utilization of a trunk group is measured using average CCS per trunk in an hour, where CCS is centi-call seconds. For example, there are 36CCS in an hour in a single trunk. If there were 100 trunks in a trunk group and all trunks were occupied for the entire hour, the average CCS per trunk for that trunk group would be 36CCS. Note that this value of 36CCS per trunk is a theoretical maximum and trunk group utilization rates are unlikely to reach that value.

In Figure 13, the average CCS per trunk for the trunk reservation network is less than that of the no-reservation network. This is because the no-reservation network allows for multi-link calls, which provides for higher utilization levels, regardless of the congestion level of the network. However, the lower average CCS per trunk for the trunk reservation network also suggests that there are available capacities to connect more calls through their direct paths when they enter the network.

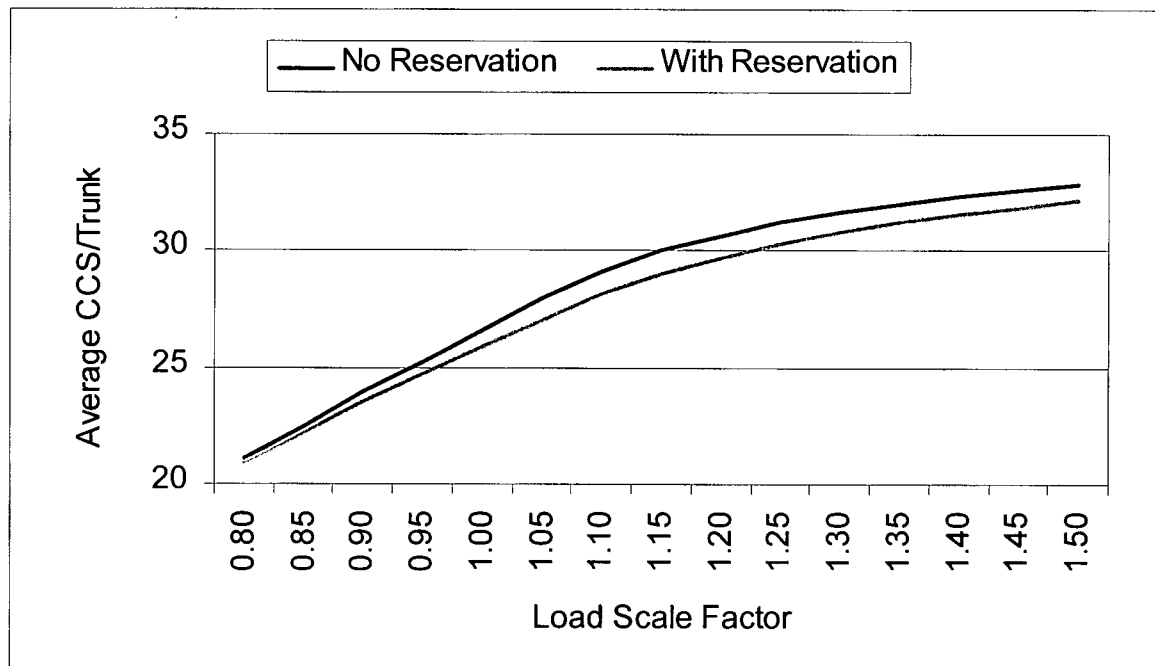


Figure 13: Average CCS per trunk for networks with and without trunk reservation at various load scale factors

As expected, Figure 14 shows that of all the calls that were connected in the network, the percentage of multi-link calls decreased drastically in the trunk reservation network when the overload level increased.

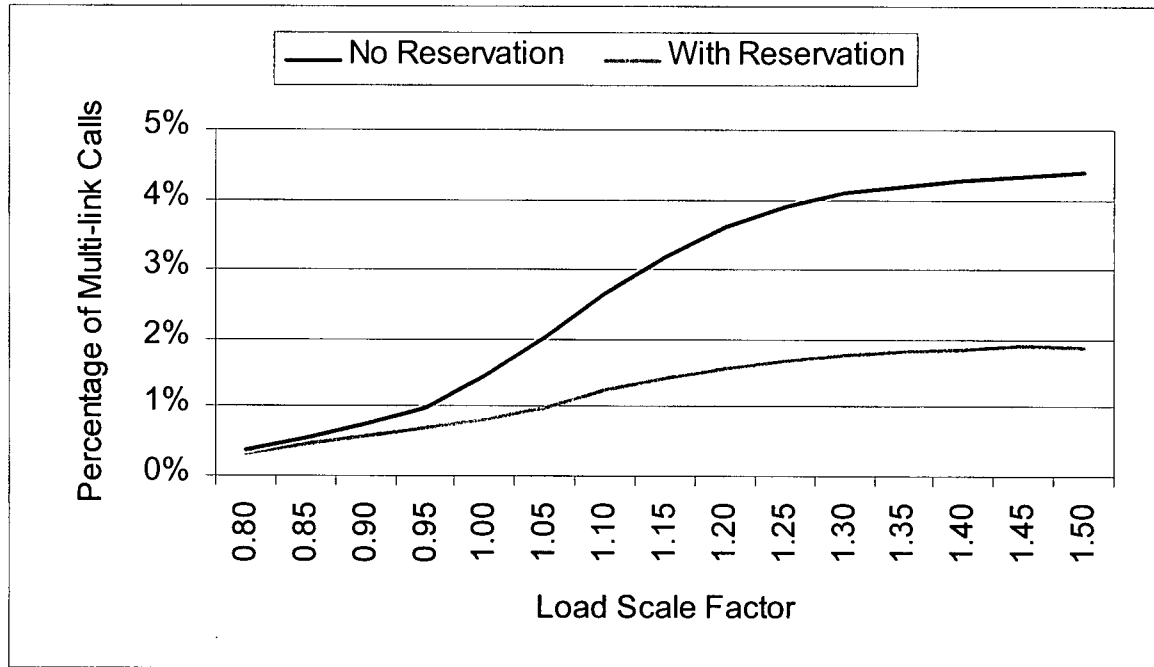


Figure 14: Percentage of multi-link calls for networks with and without trunk reservation at various load scale factors

To summarize, the simulation results show that the usage of trunk reservation is beneficial in a congested network. Trunk reservation reduces the percentage of multi-link calls in the network, which then frees up capacities to connect more calls via their direct paths leading to a more operationally efficient network.

5.4 Process to Determine the Final Optimal Reservation Parameters

A particular value for an annealing parameter is considered better than another for the same parameter if the spread (or variance) of the distribution for each trunk group is less for the first value. For example, consider two values, A and B, for an annealing parameter. In Figure 15, the variances of the distributions of the reserved trunks in the trunk groups for parameter value A are less than those of parameter value B. We can conclude, then, that parameter value A is a better annealing parameter value than B. The objective of this elimination process is to determine those values for the annealing parameters that provide small spreads, leading to good solutions. The set of results will then be pared down using these desirable values for annealing parameters to obtain the final set of trunk reservation parameters.

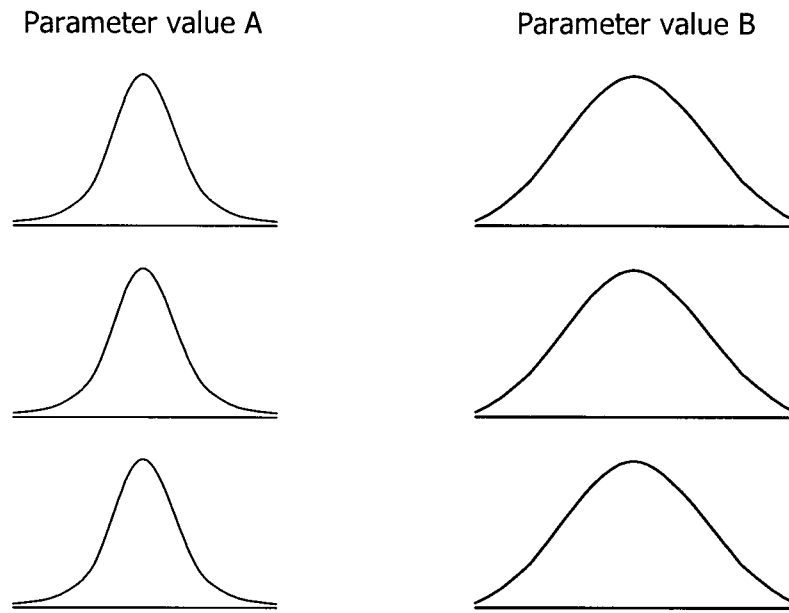


Figure 15: Illustration showing which parameter value is better

A particular value for an annealing parameter is considered better than another for the same parameter if the spread (or variance) of the distribution for each trunk group is less for the first value. For example, consider two values, A and B, for an annealing parameter. In Figure 15, the variances of the distributions of the reserved trunks in the trunk groups for parameter value A are less than those of parameter value B. We can conclude, then, that parameter value A is a better annealing parameter value than B. The objective of this elimination process is to determine those values for the annealing parameters that provide small spreads, leading to good solutions. The set of results will then be pared down using these desirable values for annealing parameters to obtain the final set of trunk reservation parameters.

5.4.1 Initial Temperature Acceptance Rate

We first take the results from all the annealing simulations using the initial temperature acceptance rate of 80%. In this filtered data set, we calculate the spread (or variance) of the number of reserved trunks for each of the 66 trunk groups. The average and standard deviation of the spreads are finally calculated. We repeat this process for the other initial temperature acceptance rates of 85% and 90%. A particular value of the initial temperature reduction rate is better than another if both its average and standard

deviation of its spread is lower. Low average and standard deviation values for the spread indicates tighter distributions for the suggested number of reserved trunks in each trunk group.

Table 3 shows the values for the averages and standard deviations of the variances for each initial temperature acceptance rate and Figure 16 provides a pictorial representation.

Initial Temperature Acceptance Rate	80%	85%	90%
Average	22.60	21.42	19.59
Standard Deviation	39.30	39.62	38.01

Table 3: Average and standard deviations of the spreads for different initial temperature acceptance rates

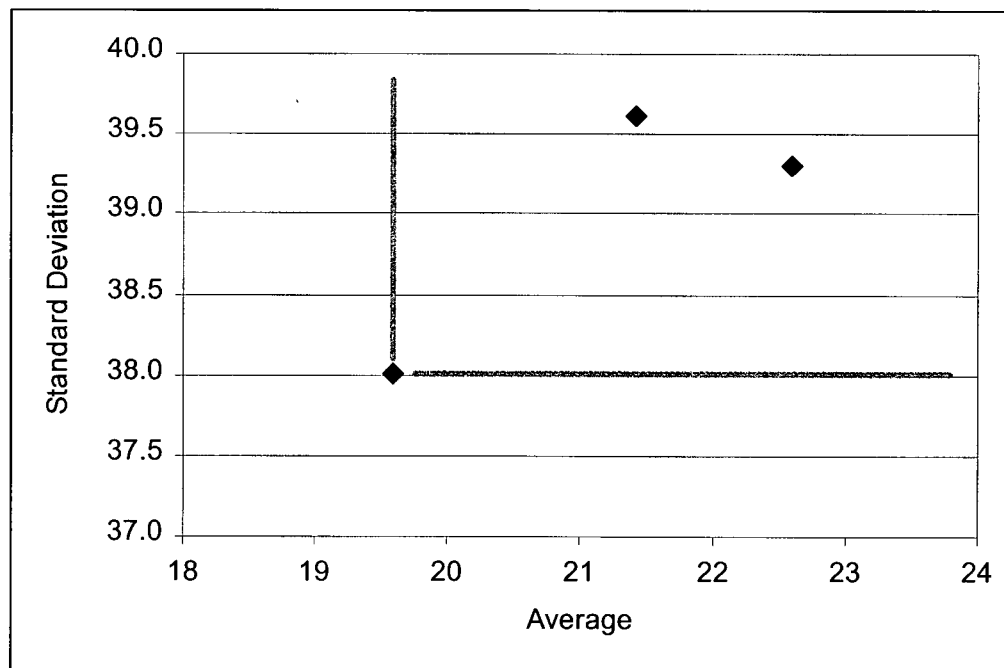


Figure 16: Plot of the averages and standard deviations of the spreads for different initial temperature acceptance rates

From Figure 16, the initial temperature acceptance rate of 90% dominates the other two rates on the average and standard deviation of the spread. Therefore, we will use only the results from annealing simulations using initial temperature acceptance rates of 90% to determine the final trunk reservation parameters.

5.4.2 Reduction Rate

Following the same procedure described in Section 5.4.1, the averages and standard deviations for the reduction rates from 40% to 90% (increments of 5%) are calculated. Table 4 shows the results of these calculations.

Reduction Rate	Average	Standard Deviation
40	15.62	38.57
45	14.29	25.80
50	14.17	23.65
55	16.07	41.39
60	15.27	28.38
65	14.12	16.33
70	17.11	25.22
75	20.11	34.57
80	24.85	35.38
85	24.72	23.06
90	37.37	50.83

Table 4: Average and standard deviations of the spreads for different reduction rates

From Figure 17, the average-standard deviation point (14.12, 16.33) corresponding to a reduction rate of 65% dominates all the other reduction rates. Therefore, we will only use the results from all annealing simulations using a reduction rate of 65% to determine the final trunk reservation parameters.

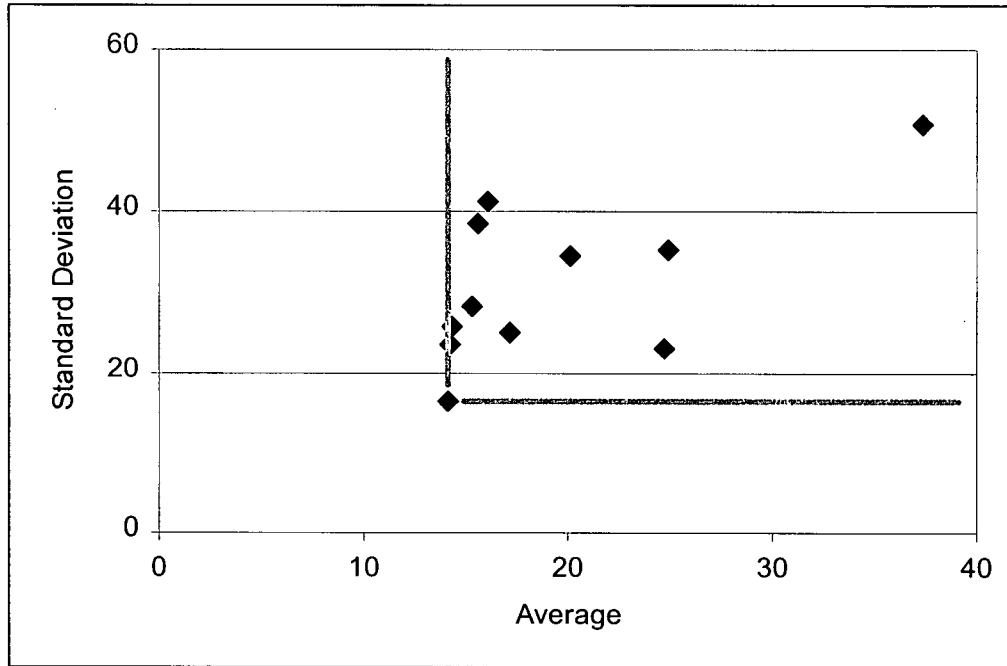


Figure 17: Plot of the averages and standard deviations of the spreads for different reduction rates

5.4.3 Acceptance Rate

Again, the averages and standard deviations of the spreads are calculated for the five different acceptance rates. Table 5 shows the results of these calculations.

Acceptance Rate	1	2	3	4	5
Average	12.39	15.07	13.53	13.73	12.94
Standard Deviation	24.53	37.21	36.46	30.89	25.98

Table 5: Averages and standard deviations of the spreads for different acceptance rates

From Figure 18, the 1% acceptance rate dominates the other four acceptance rates. Therefore, we will only use the results from all annealing simulations using an acceptance rate of 1% to determine the final trunk reservation parameters.

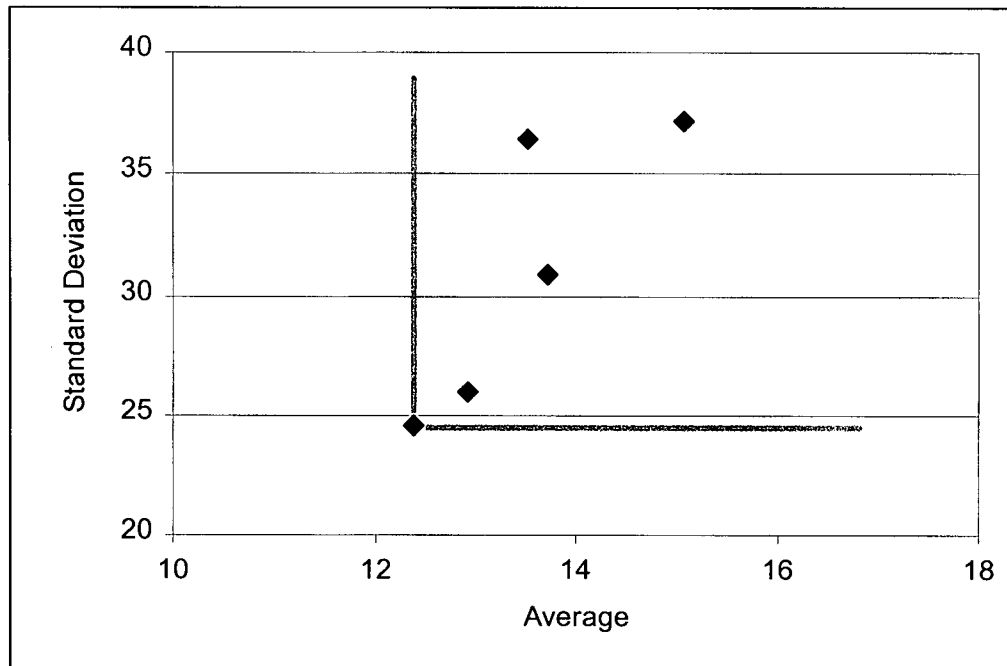


Figure 18: Plot of the averages and standard deviations of the spreads for different acceptance rates

5.4.4 Final Reservation Parameters

The optimal annealing parameter values derived from the analyses in Sections 5.4.1 to 5.4.3 are:

- an initial temperature acceptance rate of 90%;
- a reduction rate of 65%; and,
- an acceptance rate of 1%.

This combination of annealing parameters did converge to the same optimal total blocking level of 10.2751. The number of reserved trunks for each trunk group from the annealing process using the above specifications for the annealing parameters are provided in Appendix 1.

Chapter 6: Conclusions

6.1 Study Objectives

The purpose of this study was to determine trunk specific reservation parameters for a non-symmetric network with non-uniform demand. Previous studies on the use of trunk reservation focused solely on using uniform trunk reservation parameters in symmetric networks with uniform demand. Since our application network is neither symmetric in its configuration or demand patterns, the use of uniform trunk reservation parameters was felt to be inadequate. Hence, our motivation for this study to determine non-uniform trunk specific reservation parameters.

In this thesis, we showed that trunk reservation is indeed important when the network is in overload levels. It is extremely beneficial in situations when the network operates in a short-term high congestion period such as Mother's Day, for example. Trunk reservation helps mitigate against an increase in blocked calls during these periods by "increasing" capacities in the network and reducing inefficiencies by decreasing the amount of multi-link calls in the network, which translates to more calls being routed directly.

6.2 Future Studies

However, the results from simulating the optimal reservation parameters obtained from the fixed-point approximations show that the no-reservation network outperformed the optimal reservation network in terms of the total overflow level. This could be partly due to the use of the decomposition assumption in the modeling of the trunk reservation problem in the fixed-point model. This decomposition assumption has proved to be unrealistic since the results from the theoretical fixed-point model and the simulation contradicted each other. Therefore, other approaches should be considered to solve this trunk reservation problem such as semi-Markov chains.

We used the simulated annealing algorithm in this study to determine the optimal trunk reservation parameters. However, we obtained multiple optimal solutions from all

possible combinations of the three annealing parameters. We chose the final optimal reservation parameters in an ad hoc manner, and these parameters may not even be the best or in the top 10% of the "best" solutions. Further studies on using other optimization algorithms to solve this problem are required to find the globally optimal solution, or to further support the results from using the simulated annealing algorithm.

This study is one of several done by the Centre for Operations Excellence (COE) in collaboration with TELUS to provide recommendations on different and more efficient ways of operating TELUS' telecommunication networks. Two previous studies included designing optimal routing paths for calls and determining the optimal dimensions (configuration of trunks) of the networks given the demand on the network. Optimally designing an efficient network with trunk reservation to route calls using minimal dimensioning requirements is a difficult and complex task. Trunk reservation, call routing and network dimensioning are all interrelated, and any change to the parameters of either of these three modules affects the other two. This is because the parameters from the other two modules are used as inputs in optimizing the parameters of the third module. As the COE now has the knowledge base to optimize the three individual modules, the next step is to create a model that will optimize all three modules together.

Appendix 1: Final Optimal Trunk Reservation Parameters

Originating Switch	Destination Switch	Number of Reserved Trunks
0	1	37
0	2	0
0	3	0
0	4	0
0	5	0
0	6	1
0	7	0
0	8	0
0	9	0
0	10	0
0	11	2
1	2	0
1	3	0
1	4	0
1	5	0
1	6	12
1	7	0
1	8	5
1	9	3
1	10	0
1	11	3
2	3	0
2	4	2
2	5	9
2	6	0
2	7	0
2	8	1
2	9	4
2	10	0
2	11	0

Originating Switch	Destination Switch	Number of Reserved Trunks
3	4	9
3	5	4
3	6	0
3	7	8
3	8	5
3	9	3
3	10	8
3	11	2
4	5	0
4	6	10
4	7	5
4	8	7
4	9	1
4	10	0
4	11	0
5	6	5
5	7	6
5	8	7
5	9	7
5	10	8
5	11	2
6	7	0
6	8	13
6	9	5
6	10	4
6	11	1
7	8	1
7	9	7
7	10	1
7	11	9

Originating Switch	Destination Switch	Number of Reserved Trunks
8	9	3
8	10	3
8	11	0
9	10	5
9	11	0
10	11	7

References

- Aarts EHL, Korst J. *Simulated Annealing and Boltzmann Machines*, Wiley Inter-Science Publication, Essex, 1989.
- Akinpelu JM. The Overload Performance of Engineered Networks with Nonhierarchical and Hierarchical Routing. *AT&T Bell Labs Tech. Journal* 63 (Sept 1984) 1261-1281.
- Ash GR, Cardwell RH, Murray RP. Design and Optimization of Networks With Dynamic Routing. *Bell System Technical Journal* 60 (Oct 1981) 1787-1820.
- Baird, S. Personal conversation. 2001
- Beasley JE, Chu PC. A genetic algorithm for the set covering problem, *European Journal of Operational Research* 94 (1996) 392-404.
- Cerny V. A Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm, *Journal of Optimization Theory and Applications* 45 (1985) 41-51.
- Chen, H. Personal conversation. *Professor of Operations and Logistics at University of British Columbia*. 2001
- CISCO SS7 PRI Gateway Solution for Accelerating Packet Profits*. (2000) CISCO Systems. Retrieved August 8, 2001 from the World Wide Web:
http://www.cisco.com/warp/public/cc/pd/si/mg8260/prodlit/sprig_an.htm
- Franks RL, Rishel RW, Overload Model of Telephone Operation. *Bell System Technical Journal* 52 (Nov 1973) 1589-1615.
- Gibbons RJ, Kelly FP. Dynamic routing in fully connected networks. *IMA Journal of Mathematical Control and Information* 7 (1990) 77-111.
- Glover F. Tabu Search: Part I, *ORSA Journal on Computing* 1 (1989) 190-206.
- Glover F. Tabu Search: Part II, *ORSA Journal on Computing* 2 (1990) 4-32.
- Goldberg DE. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, 1989.
- Johnson DS, Aragon CR, McGeoch LA, Schevon C. Optimization by Simulated Annealing: An Experimental Evaluation; Part 1, Graph Partitioning. *Operations Research* 37 (1989) 865 – 892.
- Kabanuk, S. *Nonhierarchical Alternate Routing in a Circuit-Switched Network at TELUS*. Masters thesis, University of British Columbia (2000).

- Kelly FP. Routing in Circuit-Switched Networks: Optimization, Shadow Prices and Decentralization. *Advances in Applied Probability* 20 (1988) 112-144.
- Kelly FP. Routing and Capacity Allocation in Networks with Trunk Reservation. *Mathematics of Operations Research* 15 (Nov 1990) 771-792.
- Kirkpatrick S, Gelatt CD Jr., Vecchi MP. Optimization by Simulated Annealing, *Science* 220 (1983) 671-680.
- Lippman SA. Applying a New Device in the Optimization of Exponential Queueing Systems. *Operations Research* 23 (1975) 686-710.
- Metropolis N, Rosenbluth A, Rosenbluth H, Teller A, Teller E. Equations of State Calculations by Fast Computing Machines, *Journal of Chemical Physics* 21 (1953) 1087-1091.
- Miller BL. A Queueing Reward System with Several Customer Classes. *Management Science* 16 (1969) 234-245.
- Pirlot, M. General Local Search Methods, *European Journal of Operational Research* 92 (Aug 1996) 493-511.
- Reeves CR. *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific, 1993.
- Ross, SM. Stochastic Processes (Second Edition). *Wiley Series in Probability and Mathematical Statistics*. John Wiley & Sons, 1996.
- Simons, R. Modelling Telecommunications Networks. *MP in Action* (Jan 1997), Eudoxus Systems Ltd. <http://www.eudoxus.com/mpac9701.pdf> (Retrieved July 17, 2001)
- Smith, I. *An Application of Multivariate Analysis to Time of Day Routing in Telecommunication Networks*. Masters thesis, University of British Columbia (2000).
- TELUS Investor Fact Sheet*. (Mar 2001). TELUS Corporation. Retrieved July 16, 2001 from the World Wide Web: <http://about.telus.com/downloads/1Q2001-factsheet.pdf>
- van Laarhoven PJM, Aarts EHL. *Simulated Annealing: Theory and Applications*, D Reidel Publishing, Holland, 1987.
- Washburn, B. (1996). *The Telecom Industry in Canada* [Electronic Version] Available online at <http://www.telecoms-mag.com/misc/CanadaSupplement/c-wash.html>. Downloaded July 11, 2001.
- Wolsey, LA. *Integer Programming*, Wiley Inter-Science Publication, New York, 1998.