Spring 1998

A combined simulated annealing and TABU search strategy to solve network design problem with two classes of users

Qifeng Zeng
New Jersey Institute of Technology

ABSTRACT

A COMBINED SIMULATED ANNEALING AND TABU SEARCH STRATEGY TO SOLVE A NETWORK DESIGN PROBLEM WITH TWO CLASSES OF USERS

by
Qifeng Zeng

A methodology to solve a transportation network design problem (TCNDP) with two classes of users (passenger cars and trucks) is developed. Given an existing highway system, with a capital investment budget constraint, the methodology selects the best links to be expanded by an extra lane by considering one of three types of traffic operations: exclusive for passenger cars, exclusive for trucks, and for both passenger cars and trucks such that the network total user equilibrium (UE) travel time is minimized.

The problem is formulated as an NP-hard combinatorial nonlinear integer programming problem. The classical branch and bound methodology for the integer programming problem is very inefficient in solving this computationally hard problem. A combined simulated annealing and tabu search strategy (SA-TABU), was developed which is shown to perform in a robust and efficient manner in solving five networks ranging from 36 to 332 links. A comprehensive heuristic evaluation function (HEF), a core for the heuristic search strategy, was developed which can be adjusted to the characteristics of the problem and the search strategy used. It is composed of three

elements: the link volume to capacity ratio, the historical contribution of the link to the objective function, and a random variable which resembles the error term of the HEF.

The principal characteristics of the SA-TABU are the following: HEF, Markov chain length, "temperature" dropping rate and the tabu list length. Sensitivity analysis was conducted in identifying the best parameter values of the main components of the SA-TABU. Sufficiently "good" solutions were found in all the problems within a rather short computational time. The solution results suggest that in most of the scenarios, the shared lane option, passenger cars and trucks, was found to be the most favored selection. Expanding approximately 10% of the links, results in a very high percentage improvement ranging from 73% to 97% for the five test networks.

A COMBINED SIMULATED ANNEALING AND TABU SEARCH STRATEGY TO SOLVE A NETWORK DESIGN PROBLEM WITH TWO CLASSES OF USERS

by
Qifeng Zeng

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Transportation

Interdisciplinary Program in Transportation

May 1998

APPROVAL PAGE

A COMBINED SIMULATED ANNEALING AND TABU SEARCH STRATEGY TO SOLVE A NETWORK DESIGN PROBLEM WITH TWO CLASSES OF USERS

Qifeng Zeng

Dr. Kyriacos C. Mouskos, Dissertation Adviser — Date
Assistant Professor of Civil and Environmental Engineering, NJIT

Dr. Athanassios K. Bladikas, Committee Member — Date
Associate Professor of Industrial and Manufacturing Engineering, NJIT

Dr. Norbert Oppenheim, Committee Member — Date
Professor of Civil Engineering, City University of New York

Dr. Louis J. Pignataro, Committee Member — Date
Executive Director of Institute for Transportation, NJIT

Dr. Lazar N. Spasovic, Committee Member — Date
Associate Professor of Management, NJIT

BIOGRAPHICAL SKETCH

Author: Qifeng Zeng

Degree: Doctor of Philosophy in Transportation

Date: May 1998

Undergraduate and Graduate Education:

- Doctor of Philosophy in Transportation,
  New Jersey Institute of Technology, Newark, NJ, 1998
- Master of Science in Transportation,
  The City College of New York, New York, NY, 1992
- Bachelor of Science in Civil Engineering,
  Southeast University, NanJing, P.R.China, 1988

Major: Transportation

Presentations and Publications:

Q. Zeng and K. C. Mouskos, "Simulated Annealing Heuristic Search Strategy to Solve a Two Classes of Users Transportation Network Design Problem," Presented at *INFORMS Conference*, Dallas, Texas, Fall 1997.

K. C. Mouskos and Q. Zeng, "A Simulated Annealing Heuristic Search Strategy for the User Equilibrium Transportation Network Design Problem," Present at *INFORMS Conference*, Washington, D. C., Spring 1996.

This thesis is dedicated to
My Parent, Zeng, Xiang Jian and Zheng, Shu Xiu,
and my wife, Chan, Mei Mei

ACKNOWLEDGMENT

I would like to express my sincere appreciation to my supervisor Dr. Kyriacos C. Mouskos for his continuous support, guidance, and friendship throughout my Ph.D. study and research work.

Special thanks to Dr. Louis J. Pignataro, for his supervision and recommendation for the funding of this study, and Dr. Norbert Oppenheim, whose encouragement and recommendation brought me to NJIT Ph.D. program. I would also like to extend my appreciation to Dr. Athanassios Bladikas and Dr. Lazar N. Spasovic for their advice and supports.

Finally, I would like to extend my gratitude to my classmates in NJIT transportation program.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF TABLES
(Continued)

Table **Page**

LIST OF TABLES
(Continued)

Table **Page**

LIST OF TABLES
(Continued)

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

This dissertation presents a methodology to solve a transportation network design problem with link improvements to the existence of cars and trucks in the traffic stream. Highway safety and operational efficiency are two of the primary issues of highway capital program management. The motivation for this study stems from the disparity in the operational characteristics between trucks and passenger cars, and the unavailability of a methodology that provides a systematic way in identifying the best candidate improvements to be made on a highway network. The present methodologies applied usually consider a rather small number of alternatives without consideration to their global effect on the network.

1.1 Overview of Current Urban Transportation System and Network Design Problem

In recent decades, most of the major urban transportation systems in United States have been characterized by roadway congestion. Congestion causes longer commuting time, higher vehicle operating cost, and consequently brings several social and economical issues to the stage by impeding on the regional economical development, excelling air pollution and impacting the residents' daily life qualities. The trend in staggering travel demand and the inadequacy of the existing transportation facility capacity further sharpens this problem.

Various efforts have been undertaken to alleviate the urban transportation system congestion. To mention some, Intelligent Transportation System (ITS) uses advanced communication and information systems to improve highway mobility; high occupancy vehicle (HOV) lane policies increase highway passenger capacity by carrying more passengers; improvements of transit systems attract more automobile riders to high capacity transit systems boosting the transportation system's overall flow throughputs.

Due to the scarcity of available capital and the low cost-benefit value of large capital investments for new infrastructure, today, most of the urban transportation capital investment projects are focused in maintaining and improving the current transportation facilities. In optimizing the limited capital resources, one of the most widely used methods is the implementation of project cost and user benefits analysis. The transportation network design problem is one of the tools for conducting such types of transportation network investment analysis.

The transportation network design problem is an integral part of the transportation planning process. It provides answers to the question: how to optimize the use of the capital investments in an existing facility based on a set of objectives and subject to specific constraints of the specific problem. Furthermore, recent developments in transportation system analysis have assigned a broader meaning to the network design problem, such as: toll policy, specifying traffic direction in certain streets, ramp metering and setting exclusive passenger car lane, traffic control improvements, HOV lane and other types of operational improvement options.

The main functionality of the transportation network is to move goods and people. In the highway system, such movement is presented by various types of vehicles operating on the roadways. The principal differences among these vehicles are their sizes and operational characteristics. In general, passenger cars and heavy trucks are two primary aggregated classes for the traffic stream. The passenger car is usually a small vehicle with 30 to 60 lb/hp weight-to horsepower ratio. The heavy trucks can have more than 200 lb/hp weight-to-horsepower ratio and are more than double and triple the size of regular passenger cars. It is quite significant to consider passenger cars and trucks as two different classes of users in the highway network analysis. Compared with the passenger car, the truck has bigger size while demanding more roadway space, and its "Crawl speed" is much more lower when traveling over a significant distance at over 2% grade. Thus, the heavy truck induces much more impact on the roadway capacity than the regular passenger car. In the 1994 Highway Capacity Manual, passenger car equivalent factors are used to capture the cars and trucks disparity on the roadway capacity, where the truck's impact is considered to be three to four times higher than the passenger car at low grades and can be as high as 28 for high grades. Additionally, the trucks have a much more severe impact on the roadway pavement life period.

In recent decades, the change in traffic composition has been characterized by a growing proportion of heavy trucks in the traffic stream and an increase to the traffic operational difference between the passenger cars and trucks. Cars are becoming smaller, lighter and less powerful, while trucks are becoming larger and more powerful. It raises the issue of efficiency and safety of the roadway where two different vehicles in terms of size, speed and acceleration rate, competing for the same right of the way, and share the same facility simultaneously. The truck accidents in California have increased by 10% per year since 1985, while their total number was 12,000. A delay average of 2,500 vehicle-hours have been noted for each truck related accident in California (Reference from Middleton (1996)).

Some efforts in addressing the passenger cars and trucks dissimilarity have been under taken in improving the highway efficiency and safety. For example, climbing lane are utilized for slow trucks on certain long uphill roads; the trucks are prohibited in the left most lane in some multilane highways (freeways); and in some heavy duty highway networks, such as the New Jersey Turnpike, exclusive passenger car roadway and shared roadway networks are used to separate passenger cars and trucks.

1.2 Overview of the Problem Structure and Methodology

The core of this study is a network design problem where the roadway capacity is expanded by building an extra lane on selected links which may be associated with any one of three types of traffic operation- both passenger cars and trucks allowed, exclusively for trucks or exclusively for passenger cars. The primary issues that this study focuses are the following: First, given a specific network configuration and the two origin-destination (O-D) matrices - the passenger car O-D matrix, and the truck O-D matrix, how are the flows distributed on the links of the network? Second, what is the optimal network configuration based on the three alternatives mentioned above, that will minimize the network total travel time, subject to a budget constraint on the number of allowable lane additions? The problem is subsequently referred to as a two classes of users (Passenger Cars and Trucks) transportation network design problem (TCNDP).

The main objective is the total travel time of the network, which is the summation of the User Equilibrium (UE) passenger cars and trucks travel times on all the links. The most common travel time function used is the Bureau of Public Roads (BPR) travel time cost function where the link travel time is a function of the traffic flow and capacity. The links' traffic flows are assigned by the UE traffic assignment rule. Therefore the problem is classified as a two level (called bilevel) optimization problem. In addition to the objective function - minimizing the UE total travel time, the link traffic flows are obtained by solving another optimization problem - minimizing the individual traveler's travel time. One of the methodologies in solving the problem is by separating the two level optimization into two sequential procedures: first predicting traffic flows on the links according to the current network configuration and then identifying the links to be improved links to optimize the total UE network total travel time.

This problem falls into the category of nonlinear and integer programming problems. The classical method to solve such discrete transportation network design is the branch and bound method. The efficiency of the branch and bound method lies in its emphasis in the determination of good bounds or cuts to reduce solution space.

However, the size of the solution space increases exponentially with the network size, whereas the execution time experienced by the traffic assignment solution algorithm is rather slow, restricting the application of the branch and bound method to fairly small networks. In recent years, heuristic algorithms are becoming more and more promising in empirical applications with large network sizes. In a simplified view of the heuristic search strategy, a gathering of information, drawn from historical data and past experience is used to guide the move - locate the new solution state from the current solution state, until the global optimal solution state or a satisfactory solution state is found.

1.3 Motivation and Objectives

The importance of the TCNDP problem has been highlighted in Section 1.1. Due to the complexity of the problem structure and the computational difficulties, no intensive study of this problem has ever been conducted.

The current transportation planning process addressed the problem of truck lane needs on an empirical manner rather than a global optimization objective. Often, safety supersedes the decision to address truck related improvements. Traditionally, the most common truck-related improvements are observed on high grades where truck climbing lanes are proposed to improve the operation of the roadway, improving the speed of the passenger cars. Other types of truck related improvements is observed in freeways where trucks are often prohibited from using the left (high speed) lanes. Additionally, trucks are restricted to specific roadways of the network. Parkways, usually do not allow heavy vehicles to operate on them, whereas in some cases trucks are restricted during certain hours of the day. However, a systematic approach in identifying the most optimal truck related strategies to be implemented on a network-wide basis does not appear in the literature. This study presents a methodology for addressing truck lane needs on a network wide basis. The problem was first addressed by Mahmassani et. al. (1985), although no efficient optimization procedure was presented that time. The advancements in computational efficiency of computer hardware and recent advancements in the application of heuristic search strategies provided the motivation for re-examining the original problem. The primary objectives of this dissertation are presented below:

Objective 1:

Present a mathematical formulation of the transportation network design problem with two types of users, passenger cars and trucks. The problem is formulated as a bi-level mathematical program. The lower level addresses the identification of the link flows, passenger cars and trucks. The upper level addresses the truck related improvements considered in this study: i) both passenger cars and trucks allowed, ii) exclusively for trucks, iii) exclusively for passenger cars.

Objective 2:

Investigate the application of the diagonalization algorithm to solve the user equilibrium traffic assignment with user asymmetric interactions, passenger cars and trucks. A primary concern of this traffic assignment procedure is whether a unique solution exists or not.

Objective 3:

Develop a heuristic search strategy to solve the UE transportation network design problem with two classes of users. To analyze the proposed solution methodology, the following tasks were undertaken:

- Sensitivity analysis on different networks was performed that assessed the effectiveness of the search strategy.
- Sensitivity analysis on different parameters and coefficients of the search was performed to identify their "best" values in optimizing the search strategy.
- An analysis of the main features of the SA-TABU used in solving the transportation network design problem.

In summary, the complexity and the large size of the empirical network used to discourage the application of the network design problem in transportation planning practice, coupled with rather slow computing facilities and lacking of empirical application oriented methodologies. Additionally, the TCNDP problem has distinguished itself as one of the most important issue in highway planning, as well as highway safety. This study presents a heuristic search strategy based on the combination of simulated annealing and tabu search, providing a promising methodology for addressing an otherwise computationally intractable problem. Furthermore, this study provides an additional tool to highway network design and planning.

1.4 Overview of this Dissertation

This dissertation includes an introduction, a literature review, methodology development and the numerical experiments. The literature review covers the formulations and algorithms involved in transportation network design problem, the single user class and two classes of users traffic assignment formulation for the User Equilibrium (UE) and System Optimum (SO) rules, the Frank-Wolfe algorithm (Reference from Sheffi (1985)), as well as the diagonalization algorithm. As part of the solution methodology review, the simulated annealing and tabu search algorithm are presented in detail.

The primary literature review on the transportation network design problem formulation and methodology is presented in Chapter 2. Chapter 3 constructs the formulation and reviews the current methodology status for the TCNDP problem specified in this study.

In Chapter 4, a comprehensive study for the two classes of user traffic assignments and the diagonalization algorithm is presented, followed by TCNDP problem. The study includes an analysis of the mathematical formulation, numerical example analysis and sensitivity analysis study.

In Chapter 5, the SA-TABU search strategy developed is presented. A complete analysis of the search strategy and its rationales can also be found in this chapter, as well as description of the computer program.

Numerical experiments on the search strategy under different sets of the parameters and coefficients on five networks are presented in Chapter 6. The experiments evaluate the quality of the proposed search strategy and the heuristic values.

Chapter 7 presents insights derived from the numerical experiments conducted on a sample networks. It presents the characteristics of the TCNDP problem and the basic conclusion and guidelines derived from empirical application.

Chapter 8 summarizes the general conclusions from the study and provides recommendations for future studies.

CHAPTER 2

LITERATURE REVIEW

The equilibrium transportation network design problem falls into the category of integer non-linear programming problems, and various of the methodologies involved in solving the integer programming problems could be employed to solve this problem. However, the classical integer programming methodologies in solving large scale and complicated problems, have limited capabilities. Recently, heuristic search strategies such as simulated annealing, tabu search, neural network and others, are used more often to solve computationally hard integer programming problems. This chapter provides a review of transportation network design models and solution algorithms or heuristic search strategies, with emphasis given to the equilibrium transportation network design problem formulation, the simulated annealing and tabu search methodologies.

2.1 Transportation Network Design

The Transportation network design covers a broad range of issues arising in transportation planning and other related fields. Various formulations and solution algorithms are presented in the literature. One of the most comprehensive reviews on transportation network design can be found in Magnanti and Wong (1984).

The basic network components are arcs, nodes and centroids. In a network, there usually exist a vector of commodities to be transported from some centroids (origin) to some centroids (destination). Thus, the general network design problem can be addressed as follows: Construct a new network with a number of links (arcs), satisfying some objective function subject to the specific constraints of the problem. Magnanti, and Wong(1984) summarized the transportation network design problems in terms of demand structures, objective functions, types of capacity and side constraints. The general model presented in this study leads to some more specific models such as the minimal spanning tree, shortest path, steiner tree problem, (nonlinear cost) multi-commodity flow problems, minimal directed spanning tree, traveling salesman problem, vehicle routing, facility location, fixed charge network design problem, network design traffic equilibrium, and budget problems.

Solution methodologies discussed in Magnanti, Wong(1984) includes Benders decomposition, branch and bound, Lagrangean relaxation, linear programming and heuristics method. Despite the differences in each approach, most of the algorithms can be categorized as: i) identifying good constraints (cuts) such as the Bender cuts; ii) approximating the current problem by one where a solution methodology is known to be more efficient or iii) searching for a non-optimum but sufficiently good solution using heuristics rule.

The UE transportation network design is one of selecting a set of network links to optimize the objective(s) (UE total travel time) by changing the selected links' capacities, while the network flow distribution follows the UE traffic assignment rule. The UE traffic assignment rule best describes the users' behavior, where at equilibrium no traveler can improve its travel cost by unilaterally changing routes (Wardrop 1952).

The selection of the solution methodology for a specific network design problem usually depends upon the network configuration and problem formulations. Although there exist various network design formulations, these formulations differ in the following aspects: i) the nature of the origin-destination matrix (i.e., single class or multiple classes of users trips); ii) link cost function (i.e., capacity constraint); iii) form of the objective function (i.e., single or multiple objectives, linear or nonlinear); iv) the constraints (i.e., linear or nonlinear); v) the traffic assignment rules (i.e., user equilibrium(UE), system optimum(SO), all or nothing); vi) the design variables data structure (i.e., discrete or continuous).

Following, some representative network design algorithms and applications, as well as some key issues concerning the solution methodology and problem formulation in the literature, are presented:

2.1.1 Bilevel Programming

The solution to an equilibrium transportation network design problem involves two procedures: the first predicts the flow distribution on the links of the network with a given demand, and the second optimizes the objective function by defining the design variables. Therefore, the problem is a two level programming problem, which falls under the category of bilevel programming.

In this study, the decision variables determine the links to be expanded or not, thereby defining a new network configuration. The objective is to minimize the total network UE travel time, which is a function of the decision variables and the link flow pattern. The link flow pattern can be obtained by solving another nonlinear optimization problem - minimizing the user's individual travel time, based on the current network configuration, which is defined by the decision variables (e.g. to build or not to build an extra lane). The solution of these two vectors of unknown variables is required to satisfy two objective functions simultaneously.

According to Bard (1983), the general formulation form of bilevel programming problems is as follows:

$$\text{MIN } F(x,y) \tag{2.1.1}$$

where x is the optimum solution for:

$$\text{MIN } f(x,y) \tag{2.1.2}$$

with the constraint:

$$G(x,y) > b \tag{2.1.3}$$

The objective function MIN F(x,y) is referred to as the upper level problem, and MIN f(x,y) for fixed y as the lower level problem. In this study, the variables y in the upper level are the variables referring to the network configuration components, (adding or not adding a new lane for either exclusively for cars, trucks or both), and the lower level variables x are the link traffic flows.

It is well documented that the bilevel programming problem is an NP-hard (Anandalingam, et. al , 1992 and Ben-Ayed, et. al., 1980). Bard(1983) proposed an problem approximate approach to solve the above bilevel problem. In his procedure, a substitute objective function is defined as a convex combination of the upper and lower objective functions:

$$P(x,y;\lambda) \equiv \lambda[F(x,y)] + (1-\lambda)[f(x,y)] \tag{2.1.4}$$

where λ is a fixed parameter.

To solve the UE transportation network design problem, after setting $\lambda = 1$, the iteration index $k = 1$ and choosing the tolerance $\varepsilon > 0$, Bard's bilevel programming algorithm becomes:

STEP 1. Minimize (2.1.4) subject to (2.1.3) and denote the solution by (x^k, y^k).

STEP 2. Check whether the link flows x^k are user optimum flows. If so, stop; the solution (x^k, y^k) is an optimal solution to the network design model.

STEP 3. Using sensitivity analysis on the objective function coefficient λ, find $\lambda_{\min} \geq 0$, the smallest value of λ for which (x^k, y^k) remains optimum in the program. Change λ to $\lambda_{\min} - \varepsilon$ and go to STEP 1.

Bard's procedure terminates in a finite number of steps. The solution obtained in the first step has the parameter value $\lambda = 1$, and, in fact, this is the network design model with the SO flow, since the UE flow is weighted to be zero. As λ decreases, the weight of SO flow increases and eventually the solution approaches to include UE flow only.

LeBlanc and Boyce (1985) indicated that if formula (2.1.2) can be solved by the Frank-Wolfe algorithm, the constraint (2.1.3) for the combined model (2.1.4) is redundant, for that the Frank-Wolfe algorithm employs an all-or-nothing assignment which satisfies constraint (2.1.3). Consequently, formula (2.1.4) can be solved very efficiently using the Frank-Wolfe algorithm. Nevertheless, the formulation and the solution methodology was developed for the continuous network design problem.

However, the bilevel programming algorithm by LeBlanc et. al. (1985) is only capable of solving small size problems with simplified variables and constraints. As the network size becomes significant, and the problem becomes more complicated, this methodology is computationally prohibitive.

2.1.2 Branch and Bound Algorithm

Branch and bound is one of the most classical methodologies in solving integer programming problems. By defining the proper bounds, the algorithm reduces the search space of the feasible solution set. The procedure continues iteratively and explores the reduced feasible region only. The procedure terminates when an exhaustive search is conducted and the optimal solution is identified.

LeBlanc (1975) presented a branch and bound algorithm solution methodology for the discrete equilibrium transportation network design problem. To avoid Braess' "paradox", LeBlanc used both UE and SO rules to define lower and upper bounds. The primary concept of his approach is that given the same network configuration, the UE traffic assignment has a total travel time greater than or equal to the SO traffic assignment total travel time, and furthermore, the network with additional new links or links with expanded capacity leads to smaller SO total travel time. This solution methodology becomes rapidly intractable as the number of variables increases. It is therefore only applicable for very small networks.

Chen et. al. (1991) proposed a branch and bound with a stochastic incremental traffic assignment approach for the single class network design problem. The branch and bound method was used without taking into consideration Braess' "paradox" and the logit based incremental traffic assignment is used to reduce the number of iterations in the assignment process. The results imply, that if the network congestion increases, the algorithm may not work properly due to the weakness of the incremental method.

The success of the branch and bound approach is restricted to small size networks. In large network problems, where the solution search space grows exponentially with the scale of the network, the approach becomes computationally prohibitive. In the worst case, given a network design problem with n binary variables, it would require the solution of 2^n number of solutions of the traffic assignment routine.

Another approach involves simplifying either the problem formulation or the network representation. The network aggregation method condenses a given network into one that is small enough to be managed efficiently and effectively while still preserving some desired characteristics or satisfying certain objectives. There are two main approaches: network element extraction and network element abstraction - deletion and aggregation of insignificant network elements. Haghani et. al.(1984) used both methods on a 60 link network and the computational time is greatly improved. A similar application in the literature can be found in Chan (1976). Poorzahedy et. al. (1982) used an approximate problem to substitute the original problem, where the branch and bound methodology was employed. Though the approximation method appears quite reasonable in some applications, the optimal solution to the original problem is not guaranteed. Network aggregation suffers from the potential occurrence of Braess "Paradox". It is not known a priori which links can be grouped together or deleted without causing an increase in the network travel time. Furthermore, the better solutions may be undiscovered which can be found in the original network configuration.

2.1.3 Heuristics Search Strategies and Network Design

In recent years, the applications of heuristic search strategies to transportation network design problems attracted more attention. Unlike the classical branch and bound method, the heuristic search strategy methods aim to find sufficiently "good" solutions instead of the optimum solution and are compromising practical approaches for large scale network design problems.

A heuristic search strategy methodology, is one that solves the problem using trial and error, and is highly dependent on past experience or information about the structure of the problem. Starting from one feasible solution, a heuristic search strategy leads to another feasible solution using a certain guideline. The guideline is designed in such a way that sufficiently "good" solutions, not necessary an optimum solution, would appear during the search over a partial solution set. The solution set structure plays important role in the success of the heuristics search application.

Pearman (1979), after investigating the structure of the solution set in the network optimization problems with various optimization functions, came with the following findings: "Firstly, none of the spatial combination problems examined indicates a distribution skewed in such way that would imply the existence of a small number of very good candidates markedly superior to the main body of solution. Secondly, the road network optimization problem appears to be the most favorably disposed of all the problems analyzed from the point of view of possessing large numbers of good sub-optimal solutions. That is, it has the most positively skewed distribution of objective function values," that implies "a reasonable good solution could be found in a relatively short time." There is no doubt that this finding encourages the application of a heuristic search strategy in the transportation network design problem in terms of the solution set structure.

The most distinguished heuristic search strategies reported in the literature include: tabu search, simulated annealing, artificial neural network and genetic algorithm.

The artificial neural network optimization method designs the "generator" or "neuron", which generates the new solution state, in such a way that new solution state has a lower objective function value. The "generator" is kept on being adjusted from the feedback of the previous solution state. Xiong and Schneider (1992) applied an artificial neural network blended with a genetic algorithm to solve the single user transportation network design problem and found that the neural network algorithm does improve the solution quality and computational efficiency in some particular problems if sufficient information is available to train the "neurons". Wei and Schonfeld (1994) employed similar neural network structure to solve multi-period network design models and the test on a very small network with three link variables, and the accuracy and efficiency of the neural network methodology was acclaimed. However, due to the immaturity of the neural network algorithm, the application is very restrictive and it also needs other methods to provide the solutions to train the "neurons".

Genetic algorithms borrow the concept of evolution. The algorithm starts from a set of solution states. In each iteration, the relative good solution states are kept to generate their "children" solution states, while the bad solution states are disposed. No applications of genetic algorithms have been applied to the transportation network design problem.

The tabu search strategy (See Glover 1986, 1989, 1990, 1993) experienced tremendous development in recent years, especially in areas such as scheduling and sequencing problem. Tabu search algorithm was first introduced to the discrete single class user equilibrium transportation network design problem by Mouskos (1991). The study showed that the optimal solutions for five small networks were found in less than 500 iterations, and good solutions were reported for 3 medium size networks.

The simulated annealing search strategy is motivated by an analogy of the optimization problem to the statistical mechanics of annealing of solids. Simulated annealing has been widely used in solving large scale combinatorial problems. In a 76 arcs traditional transportation network design problem with continuous decision variables, Friesz et. al (1990) obtained a satisfactory result by employing a simulated annealing algorithm approach. Kang (1994) also used a simulated annealing algorithm in solving a one-way street network design problem. In a 5 node by 5 node square grid network, Kang used the difference of the volume/capacity and the traffic volume as the heuristics and the near optimal solution result is found. Simulated annealing theory is presented in detail later in this chapter .

2.2 Traffic Assignment Models and Formulations

Traffic assignment is one of the core procedures in transportation network analysis. It models network user's travel behavior and predicts the link travel flow pattern with given demands. The most widely used traffic assignment models are user equilibrium (UE) and system optimal (SO) traffic assignment. This section presents single class and two classes of users traffic assignment models and their solution methodologies.

2.2.1 Wardrop (1952) Principles

The classical network traffic assignment model was born as early as 1920 in the work of Pigou (Reference from Nagureney (1993)), and was further developed by Knight (Reference from Nagureney (1993)). Wardrop (1952) proposed two principles that best described network route choice that have been widely recognized and used since then.

Wardrop stated the traveler's route choice in two principles: First Principle: The travel times of all routes actually used are equal, and less than those which would be experienced by a single vehicle on any route; Second Principle: The average travel time is minimum.

The first principle is known as the user equilibrium (UE) rule, which can be further explained as follows: no user can improve his/her travel time by unilaterally switching routes, and consequently any unused route has a higher cost than the used one (between a given O-D pair).

The second principle is referred to system optimal (SO), under which users select their routes according to what is optimum from a society point of view. Under the SO the total travel cost in the system is minimized. This principle does not necessarily generate an equilibrium flow, where the users are able to improve their individual travel time by using other routes. Thereby, in most cases, the SO solution produces a non-stable system which is not realistic, unless users are "forced" to use the designated paths.

Beckman (1956) had formulated the above Wardrop principles into mathematical optimization models. In general, these problems fall under the category of a finite-dimensional variational inequality mathematical formulations.

2.2.2 Variational Inequality Theory

Variational inequality theory is frequently utilized for the network equilibrium problems, especially in optimization problems, complementary problems and fixed point problems. In the optimization problems, both constraint and unconstrained cases can be formulated as variational inequality problems.

The finite dimensional variational inequality problem is to determine a vector $x^* \in K \subset R^n$, such that

$$F(x^*)^T \cdot (x - x^*) \geq 0, \quad \forall x \in K, \tag{2.2.1}$$

where F is a given continuous function from K to R^n and K is a given closed convex set.

One of the variational inequality theory theorems is as follows:

Let x^* be a solution to the optimization problem:

$$\text{Minimize } f(x) \tag{2.2.2}$$

$$\text{subject to: } x \in K$$

where f is continuously differentiable and K is closed and convex. Then x^* is a solution of the variational inequality problem:

$$\nabla f(x^*)^T \cdot (x - x^*) \geq 0, \quad \forall x \in K. \tag{2.2.3}$$

and vice versa. The proof of the theorem can be found in Nagurney (1993).

2.2.3 Single Class User Equilibrium Assignment Model

In the single class problem, the Wardrop (1952) first principle can be written as :

$$t_p = \begin{cases} = \lambda_w, & \text{if } x_p^* > 0 \\ \geq \lambda_w, & \text{if } x_p^* = 0 \end{cases} \tag{2.2.4}$$

where t_p is the user travel cost on the path p, x_p^* denoting the flow on the path p and λ_w is the equilibrium travel disutility associated with the Origin Destination pair ω.

Formula (2.2.4) implies that the travel time cost on the path where the flow is zero is higher than an equilibrium disutility value which is the travel time cost for the path where the flow exists. It concludes that the flow pattern generated by Wardrop's first principle is an equilibrium flow, since there is no incentive for the user to alter its traveling path.

Variational inequality governing the equilibrium condition has the following theorem: A vector $f^* \in K$, is an equilibrium pattern if and only if it satisfies the variational inequality problem

$$t(f^*)\cdot(f-f^*)\geq 0, \quad \forall f \in K. \tag{2.2.5}$$

where $t(f^*)$ is the link cost function and f is the traffic flow on the link, while K is the feasible set for the problem.

The single class UE traffic assignment then can be written as (Beckman 1956):

$$\text{MIN } z(X)=\sum_a \int_0^{x_a} t_a(w)dw \tag{2.2.6}$$

$$\sum_k f_k^{rs}=q^{rs} \qquad \forall \text{r,s} \tag{2.2.6b}$$

$$f_k^{rs}\geq 0 \qquad \forall \text{k,r,s} \tag{2.2.6c}$$

$$x_a=\sum_r\sum_s\sum_k f_k^{rs}\delta_{a,k}^{rs} \qquad \forall \text{a} \tag{2.2.6d}$$

where a denotes link a,

f_k^{rs} denotes the flow on path k for traveler from origin r to destination s,

q^{rs} denotes the total flow from origin r to destination s,

X is the link flow vector.

Constraints (2.2.6b, c, d) are the flow conservation and non negativity. The solution to the mathematical program Formula 2.2.6 is a user equilibrium flow.

The SO traffic assignment model can be directly formulated as mathematical optimization problem like problem 2.2.6. The objective function is to minimize the total network travel time, and it has the same constraints as the UE formulation. It is worth to note that the SO is quite different traffic assignment model from the UE, in spite of the similarity of their mathematical optimization formulas.

2.2.4 Two Classes User Equilibrium Assignment Mathematical Formulation

The condition (2.2.4) in the scope of the two classes of users is revised as

$$t_p^i = \begin{cases} = \lambda_w^i, & \text{if } x_p^{i*} > 0 \\ \geq \lambda_w^i, & \text{if } x_p^{i*} = 0 \end{cases} \tag{2.2.7}$$

where t_p^i is the class i user travel cost on the path p, x_p^{i*} denoting the flow on the path p for Class i, and λ_w^i is the equilibrium travel disutility for Class i associated with the origin destination pair ω.

Theorem (2.2.5) would also hold for the two classes of users problems (Nagurney 1993). However, one of the factors determining whether the variational inequality problem can be formulated to the optimization mathematical formula is the link travel cost function $t(X)$. If this function satisfies the symmetric condition (to be explained later), then solving such a variational inequality problem is equivalent to solving the optimization problem:

$$Minimize_{f \in k} \sum_{a,i} \int_0^{f_a^i} t_a^i(x)dx \qquad (2.2.8)$$

where k is the feasible set for flow f, a presenting link and i is class type.

The link's travel cost function symmetric condition, or symmetrical interaction between different classes of users, means that the marginal contribution of the Class j flow on the Class i travel cost on the Link a is same as the marginal contribution of the Class i flow on the Class j travel cost on the Link a. In mathematical terms, it can be expressed as follows (See Sheffi 1985):

$$\frac{\partial t_{ai}(x)}{\partial x_{aj}} = \frac{\partial t_{aj}(x)}{\partial x_{ai}} \quad \forall ai \neq aj \qquad (2.2.9)$$

where $t_{ai}(X)$ denotes class i travel cost function on the Link a, which is the function of Link a the flow vector X, and also x_{ai}, x_{aj} denote the flows of class i and class j on Link a respectively. Asymmetrical interaction between different classes of users would occur, when the equal sign in Formula 2.2.9 changes to an unequal sign.

The Jacobian matrix (of a vector of scalar functions) is formed by arranging the derivatives of all these functions, with respect to all the arguments, in matrix form. The Jacobian matrix of $t(X)$, which is denoted by $\nabla t(X)$, includes the partial derivatives of all the link travel times to all link flows, is

$$\nabla_x t = \begin{bmatrix} \frac{\partial t_{11}(x)}{\partial x_{11}} & \frac{\partial t_{12}(x)}{\partial x_{11}} & \cdots & \frac{\partial t_{1l}(x)}{\partial x_{11}} & \cdots \\ \frac{\partial t_{11}(x)}{\partial x_{12}} & \frac{\partial t_{12}(x)}{\partial x_{12}} & \cdots & \frac{\partial t_{1l}(x)}{\partial x_{12}} & \cdots \\ \vdots & \vdots & & \vdots & \cdot \\ \frac{\partial t_{11}(x)}{\partial x_{1l}} & \frac{\partial t_{12}(x)}{\partial x_{1l}} & \cdots & \frac{\partial t_{12}(x)}{\partial x_{1l}} & \cdots \\ \vdots & \vdots & & \vdots & \end{bmatrix} \qquad (2.2.10)$$

It can be shown that (Sheffi, 1985) if the Jacobian matrix of the link travel cost function is symmetric, an equivalent mathematical program can be found, the solution of which would satisfy the UE conditions.

However, in a number of cases, the interaction is asymmetric such as the one between trucks and passenger cars. The impact to the passenger car travel time due to the change of the truck flow is different from the impact to the truck travel time resulting from the change of the car flow. This problem is formulated as a variational inequality and can not be formulated as an equivalent mathematical programming formulation.

Several direct algorithms have been found to be successful in finding the user equilibrium solution for the two classes of users UE traffic assignment. The diagonalization algorithm is one of the most commonly known procedures that is used to solve the traffic assignment with asymmetric interactions. The diagonalization algorithm is presented in Chapter 4.

2.3 Heuristic Search Strategies

Simulated annealing and tabu search are two of the primary search strategies which have been utilized in solving computationally hard integer programming problems. They are also the principal components of the SA-TABU search strategy developed in this study in solving the TCNDP problem. This section presents a detailed review of simulated annealing and tabu search strategies.

2.3.1 Simulated Annealing

In the early 1980's, Kirkpatrick, Gelatt & Vecchi (1982, 1983) and independently Cerny (1985) introduced the concept of the physical annealing process in combinatorial optimization problem. The reason originates from the analogy between the solid annealing process and the problem of solving large scale combinatorial optimization problem.

In condensed matter physics, annealing is known as a thermal process to obtain low energy states of a solid. The process contains two basic steps. At first a solid material is heated to the temperature that it would melt in a heat booth. Then the temperature is gradually decreased until the material particles arrange themselves in the ground state of the solid. In the liquid phase, all particles of the solid arrange themselves randomly. In the ground state the particles are arranged in a highly structured lattice and the system energy is the smallest.

Back to 1953, Metropolis, Rosenbluth, Teller (1953) introduced a simple algorithm to simulate the evolution of a solid in a heat bath in the annealing process. A sequence of states of a solid are generated in the following way. Given a current state i of the solid with energy E_i, then a subsequent state j is generated by applying a perturbation mechanism which transforms the current state i into a next state j by a small distortion. The energy of the next state is E_j. If the energy difference, E_j-E_i, is less than or equal to 0, the state j is accepted as the current state. If the energy difference is greater than 0, the state j is accepted with a probability of

$$\exp(\frac{E_i - E_j}{K_b T}) \tag{2.3.1}$$

where T denotes the temperature of the heat bath and K_B is a physical constant called Boltzman constant. This simulation methodology is known as the Metropolis algorithm.

The simulated annealing algorithm is the one that applies the Metropolis algorithm to a combinatorial optimization problem. The analogy between a physical many-particles system and a combinatorial optimization problem results from the following equivalence. The solution states in a combinatorial optimization problem are equivalent to the states of a physical system, and the objective function value of a solution is equivalent to the energy of a state. The temperature controlling the whole annealing process is equivalent to a parameter that is called a control parameter in the simulated annealing algorithm.

The basic simulated annealing algorithm steps are as follows:

STEP 0 Initialize i_{start}, c_0, L_0: i_{start} is the initial solution state; c_0 a control parameter and L_0 denotes maximum number of transitions allowed under one specific control parameter. Let $i = i_{start}$.

STEP 1 Set L = 0

STEP 2 Create next solution state j from current solution state i. $f(j), f(i)$ is objective function value for the States j and i. If $\exp(\frac{f(i)-f(j)}{c_k(\text{or } c_0)}) > rand(0,1)$, then $i \leftarrow j$, and $L = L+1$.

STEP 3 If L< L_k (L_0) go to STEP2, else update c_k (c_0) and L_k(L_0).

STEP 4 If stopping criteria is satisfied, stop; else go to STEP1.

The main elements of the simulated annealing algorithm are described below:

1) The acceptance criteria : $\exp(\frac{f(i)-f(j)}{c_k(\text{or } c_0)}) > rand(0,1)$

This acceptance criteria determine whether to accept the new solution state j as the new current solution state. In most similar algorithms, such as the hill climbing method, the move from the current solution to a new solution state is made only if the new solution state has a better objective function value than the current solution state. The simulated annealing algorithm uses a probability to ascertain a move, so that it allows a certain degree of deterioration by occasionally accepting the new solution state that is inferior to the current solution state in terms of the objective function value. In doing so, the algorithm can effectively overcome the limitations of local optimality.

2) The control parameter C (Also called "temperature".)

   Besides the current solution state and the new solution state objective function, the control parameter is the only factor in the acceptance criteria. The greater the control parameter is, the easier the acceptance criteria is met. Thus the control parameter C steers the whole simulated annealing process and determines the characteristics of the move. At the beginning of the iteration process, the control parameter C is usually set to a large number so that most of the new solution states would be accepted. As the control parameter C value decreases, only a few good new solution states would be accepted resulting in a relatively small deterioration. Finally, as the value of C approaches 0, no deterioration will be accepted at all and only the finding of the better solution state would lead to the transition of the current solution state to a new solution state. This means that the simulated annealing algorithm, in contrast to local search algorithms, can escape from local optimum while still exhibiting the favorable features of a local search algorithm.

3) The control parameter updating function

   In Step 3, the control parameter C is changed by the updating function. The updating function always scales down the control parameter C values. The control parameter C decreasing rate is one of the most important factors in deciding the nature of the simulated annealing process. If the decreasing rate is too high, the process would be terminated quickly due to the stricter acceptance criteria . Thus, the so called "annealing" effect can not be achieved, instead the "quench" effect would emerge, as the desirable solution can not be obtained. If the decreasing rate is too low, the simulated annealing would require a very long process to terminate to a desirable solution state so that it becomes very inefficient or even prohibitive for large scale problems.

4) The Markov chain length and its updating function

The Markov chain length decides the cycle of the constant control parameter value. It starts at a small value and then the updating function increases its value. The simulated annealing algorithm can be analyzed by the Markov chain theory. A "trial" corresponds to a transition in Markov chain theory, and the finite set of solutions is equivalent to the set of outcomes. It can be shown that in the simulated annealing algorithm, the outcome of a trial depends only on the outcome of previous trial. It is proved by Korst (1988) that the simulated annealing guarantees asymptotically convergence to the set of globally optimal solutions under the condition that a stationary distribution is obtained at each value of C, and that the probability of finding an optimal solution increases monotonically with decreasing C. The Markov chain length and its updating function determine whether the stationary distribution can be achieved or not. Due to the nature of a practical application - a finite-time application and seeking desirable solutions instead of the global optimum solution, it usually requires a substantially long Markov chain length at every control parameter C. However, if the Markov chain length is too long, it would lead to a very inefficient algorithm.

The main features of the simulated annealing process can be outlined as follows (See Korst, 1988):

1) The probability of find the optimal solution S_{opt} is equal to 1 after a large enough number of trials k. That is: $P\{X(k) \in S_{opt}\} = 1$ (2.3.2)

2) The probability of finding an optimal solution increases monotonically with decreasing C.

3) The asymptotic behavior of the simulated annealing algorithm can be approximated in polynomial time at the cost of guaranteeing of reaching optimal solutions.

4) A finite-time implementation of the simulated annealing algorithm can be realized by generating homogeneous Markov chains of finite length for a finite sequence of descending values of the control parameter.

In summary, although the finite-time simulated annealing algorithm can not guarantee to reach the optimum solution, some features are very favorable in finding sufficiently "good " solutions, such as the asymptotic behavior of the iteration procedure and that the probability of finding an optimal solution increases monotonically with decreasing of the scheduling control parameter. To design a proper simulated annealing algorithm, not only the solution state generating mechanism should be well designed, but also, more importantly, the cooling schedule. The long homogenous Markov chain length and small decreasing rate of the "temperature"-control parameter are essential for the "good" solution, but they greatly increase the computing difficulty.

The simulated annealing algorithm has been widely applied in various combinatorial optimization problems, such as scheduling and sequence problems. Matsuo et. al.(1987) used simulated annealing on the job shop scheduling problem with the makespan as objective. Laarhoven et. al.(1992) used simulated annealing to find the minimum makespan in a job shop scheduling problem. The algorithm successfully avoid the local minimum, though it is at the cost of a large running time. Suresh et. al.(1993) applied simulated annealing on the multiobjective facility layout problem and the solutions obtained compare favorably with the best known results.

2.3.2 Tabu Search Strategy

Tabu search was developed by Glover (1986, 1989, 1990, 1993) which is briefly presented below.

Tabu search, as a heuristic search strategy approach for computationally hard optimization problems, has a broad range of applications from graph theory and matroid settings, to general pure and mixed integer programming problems. It is an adaptive procedure with the ability to make use of many other methods, such as linear programming algorithms and specialized heuristics that it directs to overcome the limitations of local optimality.

The tabu search strategy as presented in Glover (1990) is described below:

To describe the workings of tabu search, we present a combinatorial optimization problem in the following form.

$$\text{Minimize } c(x) \qquad x \in X \quad \text{in } R_n \tag{2.3.3}$$

c(x) is the objective function, x ∈ X is the constraint. We define a move s to consist of a mapping defined on a subset X(s) of X:

$$s: X(s) \rightarrow X \tag{2.3.4}$$

Associated with x ∈ X is the set S(x) which consists of those moves s ∈ S that can be applied to x; i.e., S(x) = { s ∈ S: x ∈ X(s) } (and we may thus also write X(s) = {x ∈ X : s ∈ S(x) }).

Tabu search in a simple form discloses two of its key elements: constraining the search by classifying certain of its moves as forbidden, and freeing the search by a short term memory function that provides " strategic forgetting."

The operation of the procedure in simplified form is described as follows. A subset T of S is created whose elements are called tabu move. These elements of T are determined by utilizing historical information from the search process, extending up to t iteration in the past, where t can be fixed or variable depending on the application or stage of search. Membership in T is by means of an itemized list or by reference to a set of tabu conditions. Thus the simplified form for tabu search is:

STEP 1. Select an initial x ∈ X and let x*:=x. Set the iteration counter k=0 and begin with T empty.

STEP 2. If S(x) - T is empty, go to STEP 4; otherwise, set k:= k+1 and select sk ∈S(x) - T such that Sk(x) = OPTIMUM(s(x):s∈S(x) - T).

STEP 3. Let x:= sk(x). If c(x) < c(x*), where x* denotes the best solution currently found, let x* := x.

STEP 4. If a chosen number of iterations has elapsed either in total or since x* was last improved, or if $S(x) - T = \varnothing$ upon reaching this step directly from Step 2, stop; otherwise, update T (as subsequently identified) and return to Step 2.

The principal elements of the Tabu Search are the following:

1) Tabu lists T

The tabu list encloses the most recent moves. Its function is to prohibit certain moves for a period of time (number of iterations) to enter or exit the current solution state, to prevent cycling and avoid local optimum.

2) Heuristic Evaluation Function OPTIMUM($s(x)$:$s \in S(x) - T$)

This function is used to generate the new solution state s(x) from the current solution state x, which is assumed to be a best "move". If the new solution state improves the objective function value, it would be accepted as the new current solution state.

3) Aspiration Level

The use of an aspiration level function A(s,y), which depends on the move s and/or vector y, is one of the tabu search's important features. If the objective value c(s(y)) of a move s(y) is less than a prespecified aspiration level A(s,y), then the tabu status of the move may be overridden. The move is now defined as a solution-specific move, depending on both s and y. Each solution-specific move is characterized by a set of attributes. The aspiration level might be defined either for a collection of these or for a specific objective. Once a move passes the criterion then its tabu status is overridden.

4) Strategic oscillation

The strategic oscillation is also an important element tabu search, referring to the search stage where the moves are allowed to enter the infeasible region. The search oscillates back and forth between the feasible and infeasible solution space. Thus it provides the opportunities to select paths which otherwise might not be allowed. Strategic oscillation might also be useful for sensitivity analysis by providing a range of solutions around the constraint.

5) Intermediate and long term memory

The intermediate term memory function records features that are common to a set of best trial solutions during a particular period of the search. The search then continues, using these common features as heuristics to identify the new solutions. The long term memory function diversifies the search from the current search stage by using a heuristic which is usually generated from the search. It generally works in a manner opposite to the intermediate memory by penalizing good moves rather than rewarding them. This step might achieve an escape from a local optimal solution. Both functions are used for a short number of iterations and then the search continues with the original heuristics or evaluation criteria.

The results from various tabu search applications have been very encouraging. These applications include a job schedule problem, traveling salesman, mixed integer programming, as well as the transportation equilibrium network design problem and a variety of other discrete optimization problems. Knox and Glover (1988) demonstrated the power of a prototype tabu search method on a benchmark set of seven "small but hard" symmetric traveling salesman problems ranging from 25 to 110 cities. Laguna, Barns, and Glover (1991, 1992) developed powerful tabu search methods for two related single machine problem with linear delay penalties and sequence dependent set up costs.

Tabu search strategy has also ever been implemented successfully by Mouskos (1991) in solving a single user class transportation equilibrium network design problem, where the optimal solutions for 5 small networks were found in less than 500 iterations, and good solutions were reported for 3 medium networks.

CHAPTER 3

FORMULATION OF TWO CLASSES OF USERS EQUILIBRIUM TRANSPORTATION NETWORK DESIGN PROBLEM (TCNDP) ADDRESSED IN THIS STUDY

The TCNDP in this study is addressed as: Given a transportation network with a set of nodes N and a set of links A, a fixed O-D matrix representing passenger cars and a fixed O-D matrix for heavy trucks, link travel cost functions reflecting the interaction between passenger car flow and truck flow on the same roadway, and a set of links that can be expanded by one additional lane on the original network with the following options:

Option 1: Expand the link by one lane and allow all traffic on new lane,

Option 2: Expand the link by one lane, but allow only passenger car traffic on new lane,

Option 3: Expand the link by one lane, but only truck traffic is allowed on new lane,

then, the objective is to minimize the UE total travel time of the network, subject to the available budget.

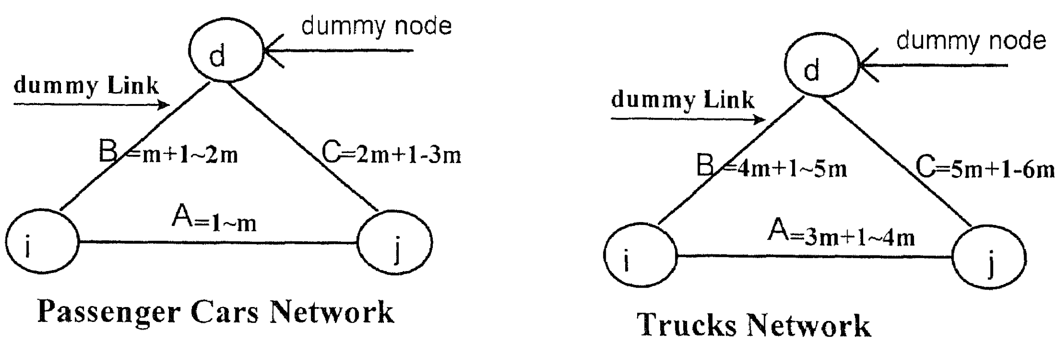
The most important issues arising in solving TCNDP are: 1) two different types of network users, passenger cars and trucks, sharing the same roadway, which possess different impacts to the roadway congestion and the interaction between themselves, and 2) the network design variable needs to include one of the three traffic operation associated with each new lane.

The TCNDP problem framework has been developed by Mahmassani et. al.(1985). Mahmassani et. al. (1985) used five options of traffic operations associated with each link and its potential additional new lane. The other two options considered are the ones that restrict the use of an existing link by the trucks in Option 1 and Option 3. The reduction to three options was made to reduce the complexity of the problem.

In order to model two different travel performance along the same roadway by passenger cars and trucks and the interactions between them, as well as the three different types of the traffic operations, a "conceptual" network presentation is used.

In the "conceptual" network presentation, each physical highway link is divided into two sub-links, one for the passenger cars and the other for the trucks. The interaction between the cars and trucks is represented by the interaction between these two sub-links. In addition, a dummy node with a dummy link and the candidate links for addition are also added on each sub-link, in order to address the three link expansion options (See Figure 3.1.1). When both the dummy links of the passenger cars and the trucks sub-links are "turned on", Option 1, the shared lane would be selected. When neither dummy links is "turned on", no build option would otherwise be selected (no expansion). Options 2 and 3 are represented by "turning on" one of the dummy links respectively. This network representation is similar to the Mahmassani, et. al. (1985) network representation.

Consider a highway network with N nodes, and assume that nodes 1,2, ..., N are passenger car nodes and N+1 , N+2, ..., 2N are truck nodes. Let passenger car sub-links (in single direction only) to be numbered 1, 2, ..., m; dummy links (defined as mechanisms to keep the existing lanes distinct from the new lanes) to be numbered



A - original highway link. Link i for cars, and links 3m+i for trucks.

B - dummy links. Links m+i for cars, and links 4m+i for trucks.

C - proposed lane additions. Links 2m+i cars, and links 5m+i for trucks.

i=1,...,m. where m= total number of existing links in single direction only

Figure 3.1.1 Example of Single Highway Link Presentation

m+1,..., 2m; and label proposed lane additions to the network 2m+1,...,3m. Thus, if i is the highway link under consideration, then m+i is the dummy link associated with Link i , and 2m+i is the lane addition associated with link i. Consequently, the truck sub-links, dummy links and proposed addition links are number from 3m+1 to 4m, 4m+1 to 5m, and 5m+1 to 6m, respectively.

The TCNDP problem mathematical formulation is customized for this study as:

$$\text{MIN} \sum_{a=1}^{6m} t_a(x_a)x_a \qquad (3.1.1)$$

subject to :

$$\sum_{a=1}^{m} c_{2m+a}y_{m+a} + \sum_{a=3m+1}^{4m} c_{2m+a}y_{m+2} \le B \qquad (3.1.2)$$

$$x_{a+m} \le My_{a+m} \quad \forall a \in (1\sim m, 3m+1\sim 4m) \qquad (3.1.3)$$

$$y_{a+m} = (0,1) \quad \forall a \in (1\sim m, 3m+1\sim 4m) \qquad (3.1.4)$$

x_a is a user equilibrium flow. (3.1.5)

where $t_a(x_a)$ is the travel time on Link a .

x_a is the flow on link a.

c_a is the fixed cost of expanding extra lane for Link a

B is the available budget

y_a is zero if Link a is not added to the network and one if included.

m is total number of existing links in single direction.

M is a very large constant.

Formula 3.1.1 is the network total travel time with respect to the user equilibrium flows; Formula 3.1.2 is the budget constrain; Formula 3.1.3 is prohibiting flow on links not constructed; Formula 3.1.4 is the decision variable definition and Formula 3.1.5 restricts the flow to be a user equilibrium flow.

Mahmassani et. al. (1984) presented a solution methodology based on the branch and bound algorithm for the TCNDP problem. However, for a binary variable 0-1 integer programming problem, the computation complexity is equal to 2^n, where n is equal to the total number of integer variables, or number of links. Consequently, the TCNDP computational complexity would grow to 4^n when the three types of traffic operation options are introduced. With presence of the high computation complexity of TCNDP problem, the branch and bound methodology is not practical even to solve a moderate small size network. Therefore, a heuristic search strategy was developed in this studyto solve problem 3.1. The developed heuristics based search strategy solution methodology for TCNDP problem in this study, which aims to find the sufficient "good" solution for middle to large size networks, is presented in Chapter 5.

CHAPTER 4

CHARACTERISTICS OF THE TRAFFIC ASSIGNMENT PROCEDURE USED IN THE TCNDP

This chapter presents a review of traffic assignment with asymmetric link interactions and the characteristics of the two classes of users equilibrium traffic assignment with passenger cars and trucks.

The primary concerns of the traffic assignment procedure are the following; 1) The form of the link travel cost function which captures the interaction between trucks and passenger cars, 2) the existence of a unique solution to the traffic assignment procedure and 3) the convergence characteristics of the solution algorithm.

4.1 Review of Traffic Assignment Algorithm

The most representative algorithms used in the single class of user traffic assignment include Frank-Wolfe algorithm and gradient projection method. The diagonalization algorithm is one of the methodologies used to solve the two classes of users with asymmetric interactions UE traffic assignment problem. In this section, the Frank-Wolfe algorithm and the diagonalization algorithm are reviewed .

4.1.1 Frank-Wolfe Algorithm (Convex Combination Method)

Since the travel cost function t_a is usually nonlinear, the UE mathematical model (2.3.6) is a nonlinear optimization problem with the linear constraints. If the travel cost function t_a is convex and differentiable in the feasible set, the general approach to such problem is to use the gradient descent method. The gradient decent method converges to the equilibrium flow. In each iteration, it takes the move along the deepest decent direction of the current state with the best step size that optimizes the current state objective function value.

The Frank-Wolfe algorithm is one of the gradient decent methods which is applicable to traffic assignment models (2.2.6). At each iteration, the Frank-Wolfe algorithm first finds a search direction by solving a linearized approximation, then solves the optimum move size along that direction. The efficiency of the algorithm derives from the fact that the direction finding step is equivalent to performing an all-or-nothing traffic assignment, that all flow between given origin and destination is assigned to the shortest path between them.

The basic steps of the Frank-Wolfe algorithm used in the UE or SO models are as below (See Sheffi 1985):

STEP 0: Initialization. Perform all-or-nothing assignment based on the free flow $t_a{}^0 = t_a^0 = t_a(0),\ \forall a.$; This yields the set of link flows $\{x_a^0\}$. Set counter n=1.

STEP 1: Update. Set $t_a^n = t_a(x_a^n),\ \ \forall a.$

STEP 2: Direction finding. Perform all-or-nothing assignment based on $\{t_a^n\}$. This yields a set of (auxiliary) link flows $\{y_a^n\}$.

STEP 3: Line search. Find optimal move size α_n that solve

$$\text{MIN} \sum_a \int_0^{x_a^n + \alpha(y_a^n - x_a^n)} t_a(w)dw \qquad \text{subject to } 0 \le \alpha_n \le 1.$$

STEP 4: Set $X_a^{n+1} = X_a^n + \alpha_n(y_a^n - X_a^n)$, $\forall$a.

STEP 5: Convergence test. If a convergence criterion is met, STOP (the current solution is the set of equilibrium link flows); otherwise, set n=n+1 and GO TO STEP 1.

4.1.2 Diagonalization Algorithm for Two Classes of Users Traffic Assignment

As described in the preceding sections, the two classes of users traffic assignment needs to capture the interaction between different classes of users. The most widely used algorithm for multiple classes of users traffic assignment procedure is the diagonalization algorithm. The diagonalization algorithm is based on solving a series of standard UE programs, which can be solved by the Frank-Wolfe algorithm. Each iteration of this procedure requires the solution of one such program. The diagonalization algorithm is presented below:

STEP 0: Initialization. Find a feasible link flow vector X^n. Set n = 0.

STEP 1 : Diagonalization. Solve the following problem:

$$\text{MIN } z^n(x) = \sum_a \int_0^{x_a} t_a(x_1^n, \ldots, x_{a-1}^n, w, x_{a+1}^n, \ldots, x_A^n)dw \qquad (4.1.2a)$$

subject to:

$$\sum_k f_k^{rs} = q^{rs} \qquad \forall r,s \qquad (4.1.2b)$$

$$f_k^{rs} \geq 0 \qquad \forall k, r, s \qquad (4.1.2c)$$

$$x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs} \qquad \forall a \qquad (4.1.2d)$$

where a denotes Link a.

x_a denotes the link flow on Link a.

$f_k^{r,s}$ denotes path k for traveler from origin r to destination s.

$q^{r,s}$ denotes the total flow from origin r to destination s.

This yields a link flow vector x^{n+1}.

STEP 2 : Convergence test. If $x^n \approx x^{n+1}$, stop. If not set n= n+1, and go to step 1.

In Sheffi (1985), it is shown that in the diagonalization algorithm, $x^n = x^{n+1}$ if and only if x^n is a vector of flows that satisfies the UE conditions, and the uniqueness of the solution requires that the Jacobian matrix (2.2.10) is positive definite. The prove of the uniqueness referred to Smith (1979) and Dafermos (1980) where a variational inequality formulation was adopted. It implies that if the algorithm converges, its solution is an equivalent flow pattern.

By noting that only the last iteration's flow pattern needs to be determined accurately, that the problem at each iteration is subject to the same set of constraints and that the solution of problem is similar to the solution of a single user class, Sheffi suggested a "streamlined" version of the diagonalization algorithm, in an attempt to reduce the computational cost. It has to be noted that the convex combinations algorithm requires many iterations to research convergence. Thus the solution of Problem 4.1.2 requires a number of internal iterations to reach convergence per outer iteration of the diagonalization algorithm. The streamlined version applies only one iteration to Problem 4.1.2, thus reducing it to a similar form as the convex-combination algorithm for a single user class (Sheffi, 1984).

The next section presents the travel cost functions used in the traffic assignment procedures of the TCNDP.

4.2 Travel Cost Functions

4.2.1 Link Travel Cost Function

This section presents brief literature review on the link travel cost functions. Both single class and two classes of users link travel cost functions, as well as the most recent travel cost functions with turn movement will be presented.

4.2.1.1 BPR Single Class Travel Cost Function: The most used link delay function in highway network projects, developed by the US Bureau of Public Roads (BPR) has the following form:

$$t_a = t_a^0(1+\alpha(\frac{x_a}{c_a})^\beta) \tag{4.2.1}$$

where t_a is the travel time on Link a.

t^0_a is the free flow on Link a.

α, β are parameters calibrated on the basis of speed limit and the link capacity.

c_a' is the "practical capacity" of Link a.

x_a is the actual flow on Link a.

Florian et. al. (1976) provided estimates for parameters α and β and the corresponding free flow travel time.

4.2.1.2 Two Classes of Users Link Travel Cost Functions: A modified BPR type travel cost function was used in Mouskos (1985) and Mahmassani et. al. (1987) to represent the interaction of passenger cars and trucks sharing the same link.

Although the function was not calibrated with any real data, it is consistent with the concept of passenger car equivalents for heavy vehicles as applied by the 1985 Highway Capacity Manual (HCM) and later the 1994 HCM. The modified version of the two classes of users travel cost function has the following form (Mahmassani, et. al., 1987)

$$t_{aA} = t_{aA}^{o}(1+\alpha_A(\frac{X_{aA}+\varepsilon\cdot X_{aT}}{C_a'})^{\beta_A}) \quad (4.2.2)$$

$$t_{aT} = t_{aT}^{o}(1+\alpha_T(\frac{X_{aA}+\varepsilon\cdot X_{aT}}{C_a'})^{\beta_T}) \quad (4.2.3)$$

Where t_{aA}, t_{aT} are the travel times of the passenger cars and trucks on Link a, respectively.

t_{aA}^{o}, t_{aT}^{o} are the free flow time of the passenger cars and trucks on Link a respectively.

α_A, β_A and α_T, β_T are roadway congestion parameters for the passenger cars and trucks respectively.

Ca' is the capacity of Link a.

X_{aA}, X_{aT} are the actual flow of the passenger cars and trucks on Link a receptively.

ε is truck equivalent factor to passenger car.

4.2.2 Link Travel Cost Functions (LTCF) Used in this Study

The two classes of users link travel cost functions used in this study are a modification of the BPR functions (See Section 2.3). The main reason for using this modified BPR curves is: first, to capture the interaction between trucks and cars, which are not present in the original curves; and second, to ensure that a unique solution can be found by solving the traffic assignment procedure.

Mahmassani and Mouskos (1987) defined the interaction between the passenger cars and trucks by converting the truck flow volume to passenger cars and derived the truck and passenger car link cost BPR functions as shown in Formula 4.2.2 and 4.2.3. However, in most of highways, the left most lanes are always passenger cars only lanes. The impact of the trucks to passenger cars consists of the indirect impact (to leftmost lane cars) and the direct impact (cars sharing same lanes). Taking into consideration into the fact that in most of highways, a high percentage of passenger cars utilize the truck free lanes to avoid the interaction with trucks, a modified link travel cost function other than the one used in Mahmassani and Mouskos(1987) which overestimates such an impact by using the direct impact over all the lanes. No study has been conducted to quantify this impact, so a moderate formula is used by scaling

Table 4.2.1 Volume/Delay Functions (Florian et.al. 1976)

Type of Roadway	Speed Limit (Mile/Hour)	α	β	Free Flow Speed(miles/hour) /Minutes Per Mile
1	0	0	1.1	
2	0-30	0.73	3.66	15.0/4.00
3	0-30	0.61	3.5	17.0/3.53
4	0-30	0.88	4.46	20.0/3.00
5	0-30	0.69	5.16	23.0/2.61
6	0-30	1.15	4.42	25.0/2.40
7	31-40	0.62	3.65	30.0/2.00
8	31-40	0.67	4.94	32.4/1.85
9	31-40	0.62	5.14	32.4/1.85
10	31-40	1.03	5.52	35.3/1.70
11	31-40	0.66	5.09	41.4/1.45
12	31-40	0.54	5.79	41.4/1.45
13	31-40	1.01	6.59	41.4/1.45
14	50	0.88	4.93	55.0/1.09
15	50	0.77	5.34	55.0/1.09
16	50	1.15	6.87	55.0/1.09

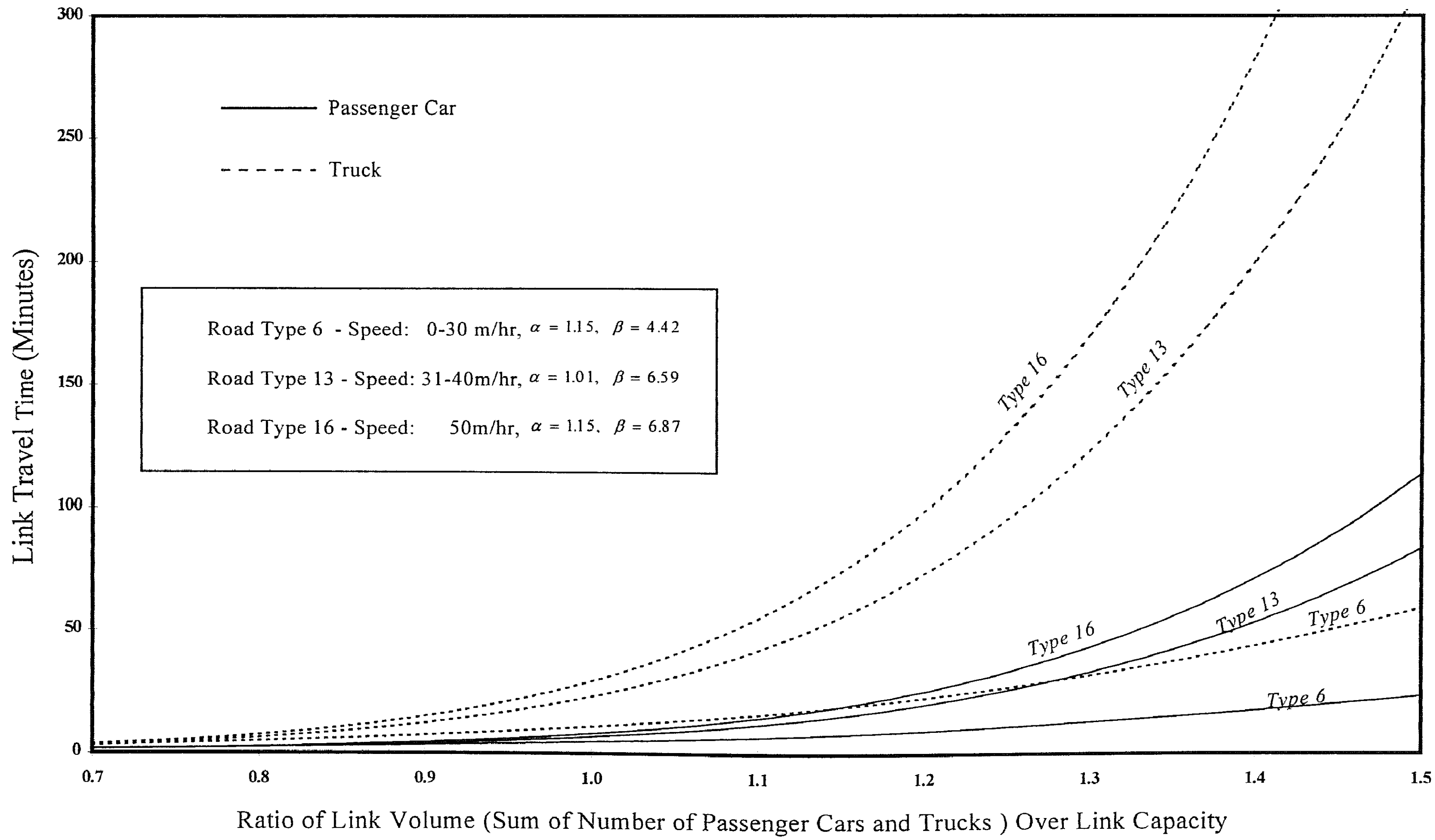


Figure 4.2.1 Two Classes of Users (Passenger Car and Truck) Link Travel Cost Function (Formula 4.2.4-4.2.5)

down the total impact of the passenger cars and trucks as the half of the direct impact. The same concept is applied to the truck travel cost function. Therefore, the modified two classes of users link travel cost function (LTCF) used in this study is formulated as:

$$t_{aA} = t_{aA}^{o}(1+\alpha_A(\frac{X_{aA}+0.5\times\varepsilon_1\cdot X_{aT}}{C_a^{;}})^{\beta_A}) \tag{4.2.4}$$

$$t_{aT} = t_{aT}^{o}(1+\alpha_T(\frac{0.5\times X_{aA}+\varepsilon_2\cdot X_{aT}}{C_a^{;}})^{\beta_T}) \tag{4.2.5}$$

The notation is the same as Formula 4.2.2 and 4.2.3. The functions are depicted in Figure 4.2.1 with the assumption that $\varepsilon_1 = \varepsilon_2$ is equal to 4; 30% of total number of vehicles are trucks, and free flow travel time for passenger cars t_{aA}^{0} is 1.0 minute and 1.2 minute for trucks t_{aT}^{0} on the link. The type of the road is consistent with the one defined in B.P.R. travel cost function (Table 4.2.1).

As mentioned in the previous chapter, the network consists of two identical sub networks, one of which is the passenger car network and the other is the truck network. The corresponding links in two sub networks share the same roadway. Thus Network G(N, A) consists of the passenger car network P(N, A), and the truck network T(N, A), Links ai (i=1, A) are the links in P(N, A) while links bi(i=1, A) is the links in T(N, A). Only the pairs of links ai and bi (i=1, n) share the same roadway. Assuming the individual link LTCF travel cost function is t_j , where j is the links in P(N, A) and T(N, A)), then the network travel cost function matrix is represented as:

$$TF = (t_{a1},\ldots t_{ai},\ldots,t_{an},t_{b1},\ldots,t_{bi},\ldots,t_{bn}) \ , \tag{4.2.6}$$

The passenger car link travel cost function is $t_{ai} = f(t_{ai}^0, \alpha_{ai}, \beta_{ai}, x_{ai}, x_{bi}, c_{ai}, \varepsilon)$ and truck link travel cost function is $t_{bi} = f(t_{bi}^0, \alpha_{bi}, \beta_{bi}, x_{bi}, x_{ai}, c_{bi}, \varepsilon)$, where t_{ai}^0, α_{ai}, β_{ai}, x_{ai}, c_{ai} and t_{bi}^0 , α_{bi}, β_{bi}, x_{bi}, c_{bi} are the free flow travel time, parameter α, β in BPR cost function, link flow and link capacity for passenger car link ai and truck link bi respectively; and ε is the truck equivalent factor to the passenger car. Since only the corresponding passenger car link and truck link share the same road, we have:

$$\frac{\partial t_{ai}}{\partial x_j} = \begin{cases} \dfrac{\partial t_{ai}}{\partial x_j} & \text{if } \ j = ai \ \text{ or } \ bi \\ 0 & \text{otherwise} \end{cases} \tag{4.2.7}$$

and

$$\frac{\partial t_{bi}}{\partial x_j} = \begin{cases} \dfrac{\partial t_{bi}}{\partial x_j} & \text{if } \ j = ai \ \text{ or } \ bi \\ 0 & \text{otherwise} \end{cases} \tag{4.2.8}$$

where i =1,..., n, ; ai=i; bi=n+i, and j=1, ..., 2n.

The corresponding Jacobian matrix, representing the pattern of link interactions, for the link travel cost functions is written as:

$$J = \begin{bmatrix} \frac{\partial t_{a1}}{\partial x_{a1}} & 0 & ...0... & ...0... & \frac{\partial t_{b1}}{\partial x_{a1}} & ...0... & ...0... & ...0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & ...0... & \frac{\partial t_{ai}}{\partial x_{ai}} & ...0... & 0 & ...0... & \frac{\partial t_{bi}}{\partial x_{ai}} & ...0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & ...0... & ...0... & \frac{\partial t_{an}}{\partial x_{an}} & 0 & ...0... & ...0... & \frac{\partial t_{bn}}{\partial x_{an}} \\ \frac{\partial t_{a1}}{\partial x_{b1}} & ...0... & ...0... & ...0... & \frac{\partial t_{b1}}{\partial x_{b1}} & ...0... & ...0... & ...0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & ...0... & \frac{\partial t_{ai}}{\partial x_{bi}} & ...0... & 0 & ...0... & \frac{\partial t_{bi}}{\partial x_{bi}} & ...0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & ...0... & ...0... & \frac{\partial t_{an}}{\partial x_{bn}} & 0 & ...0... & ...0... & \frac{\partial t_{bn}}{\partial x_{bn}} \end{bmatrix} \quad (4.2.9)$$

where

$$\frac{\partial t_{ai}}{\partial x_{ai}} = \frac{\alpha_{ai}\beta_{ai}t_{ai}^{0}}{c_i}\left(\frac{x_{ai} + 0.5\varepsilon(x_{bi})}{c_i}\right)^{\beta_{ai}-1}$$

$$\frac{\partial t_{ai}}{\partial x_{bi}} = \frac{0.5\varepsilon\alpha_{ai}\beta_{ai}t_{ai}^{0}}{c_i}\left(\frac{x_{ai} + 0.5\varepsilon(x_{bi})}{c_i}\right)^{\beta_{ai}-1}$$

$$\frac{\partial t_{bi}}{\partial x_{ai}} = \frac{0.5\alpha_{bi}\beta_{bi}t_{bi}^{0}}{c_i}\left(\frac{0.5x_{ai} + \varepsilon(x_{bi})}{c_i}\right)^{\beta_{bi}-1}$$

$$\frac{\partial t_{bi}}{\partial x_{bi}} = \frac{\varepsilon\alpha_{bi}\beta_{bi}t_{bi}^{0}}{c_i}\left(\frac{0.5x_{ai} + \varepsilon(x_{bi})}{c_i}\right)^{\beta_{bi}-1} \quad (4.2.10)$$

The Jacobian matrix is asymmetric, since $\frac{\partial t_{ai}}{\partial x_{bi}} \neq \frac{\partial t_{bi}}{\partial x_{ai}}$. As mentioned before, in this case there is no known mathematical program the solution of which is the equilibrium flow pattern, and direct approximate solution algorithms rather than a mathematical programming formulation, such as diagonalization algorithm, are used to solve the problem. However, the diagonalization algorithm requires the link-travel-time Jacobian be positive definite. Otherwise, the problem may not have unique equilibrium solution. (See Sheffi 1985).

In the presence of the asymmetric matrices, the positive definite cannot be proven by using the leading minor determinant or the eigen values. The definition for a matrix to be positive definite is presented below:

Definition: An $n \times n$ matrix A is positive definite, if and only if:

$XA(X)^T > 0$, for any X=(x_1, ..., x_i, ...x_{2n}), $x_i \subseteq$ R and one of x_i must be non zero.

The Jacobian matrix for the LTCF travel cost function is quite unique. Each row represents the impact of the volume change in one specific link to all the links. Only the link itself (passenger link) and the corresponding truck link which shares the same roadway have non-zero impact and all the others have the impact value equal to 0.

Theorem 1 : Given an $2n \times 2n$ matrix:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1n} & a_{1(n+1)} & \cdots & a_{1(n+i)} & \cdots & a_{1(2n)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & \cdots & a_{ii} & \cdots & a_{in} & a_{i(n+1)} & \cdots & a_{i(n+i)} & \cdots & a_{i(2n)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{(2n)1} & \cdots & a_{(2n)i} & \cdots & a_{(2n)n} & a_{(2n)(n+1)} & \cdots & a_{(2n)(n+i)} & \cdots & a_{(2n)(2n)} \end{bmatrix}$$

where for each row i, $a_{ii}>0$, and $a_{i(n+i)} > 0$ if i< n and $a_{i(n-i)} > 0$ if $i \geq n$, while the remaining elements are equal to zero, if

$$4a_{ii}a_{(i+n)(i+n)} > (a_{i(i+n)} + a_{(i+n)i})^2 \qquad \text{if } i < n$$
$$4a_{ii}a_{(i-n)(i-n)} > (a_{i(i-n)} + a_{(i-n)i})^2 \qquad \text{if } i \geq n$$

then the matrix A is positive definite.

Proof:

Assume any given 2n dimension vector $X=(x_1, ..., x_i, ...x_{2n})$, $x_i \subseteq R$ and one of x_i must be non zero.

$$XA(X)^T = (x_1, \dots, x_i, \dots, x_{2n}) \times$$

$$\begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1n} & a_{1(n+1)} & \cdots & a_{1(n+i)} & \cdots & a_{1(2n)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & \cdots & a_{ii} & \cdots & a_{in} & a_{i(n+1)} & \cdots & a_{i(n+i)} & \cdots & a_{i(2n)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{(2n)1} & \cdots & a_{(2n)i} & \cdots & a_{(2n)n} & a_{(2n)(n+1)} & \cdots & a_{(2n)(n+i)} & \cdots & a_{(2n)(2n)} \end{bmatrix} \times \begin{pmatrix} x_1 \\ \cdots \\ x_i \\ \cdots \\ x_{2n} \end{pmatrix}$$

Since $a_{ii}>0$, and $a_{i(n+i)} > 0$ if $i< n$, and $a_{i(n-i)} > 0$ if $i \geq n$, then:

$$XA(X)^T =$$

$$(a_{11}x_1 + a_{(n+1)1}x_{n+1}, \dots, a_{ii}x_i + a_{(n+i)i}x_{n+i}, \dots, a_{nn}x_n + a_{(2n)(n)}x_{2n}, \dots, a_{jj}x_j + a_{(j-n)j}x_{i-n},$$
$$\dots, a_{(2n)(2n)}x_{2n} + a_{(n)(2n)}x_n)$$

$$\times \begin{pmatrix} x_1 \\ \cdots \\ x_i \\ \cdots \\ x_{2n} \end{pmatrix}$$

$$= (a_{11}x_1^2 + a_{(n+1)1}x_{n+1}x_1) + \dots + (a_{ii}x_i^2 + a_{(n+i)i}x_{n+i}x_i) + \dots + (a_{nn}x_n^2 + a_{(2n)(n)}x_{2n}x_n) + \dots +$$
$$(a_{jj}x_j^2 + a_{(j-n)j}x_{i-n}x_j) + \dots + (a_{(2n)(2n)}x_{2n}^2 + a_{(n)(2n)}x_nx_{2n})$$

where $i< n$ and $j>n$.

$$= [a_{11}x_1^{\ 2} + (a_{(n+1)1} + a_{1(n+1)})x_{n+1}x_1 + a_{(n+1)(n+1)}x_n^{\ 2}] + \ldots +$$
$$[a_{ii}x_i^{\ 2} + (a_{(n+i)i} + a_{i(n+i)})x_{n+i}x_i + a_{(n+i)(n+i)}x_{n+i}^{\ 2}] + \ldots +$$
$$[a_{nn}x_n^{\ 2} + (a_{(2n)n} + a_{n(2n)})x_n x_{2n} + a_{(2n)(2n)}x_{2n}^{\ 2}]$$

$$= [(\sqrt{a_{11}}x_1 + \frac{a_{(n+1)1} + a_{1(n+1)}}{2\sqrt{a_{11}}}x_{n+1})^2 + (\frac{4a_{11}a_{(n+1)(n+1)} - (a_{(n+1)1} + a_{1(n+1)})^2}{4a_{11}})x_{n+1}^2] + \ldots +$$
$$[(\sqrt{a_{ii}}x_i + \frac{a_{(n+i)i} + a_{i(n+i)}}{2\sqrt{a_{ii}}}x_{n+i})^2 + (\frac{4a_{ii}a_{(n+i)(n+i)} - (a_{(n+i)i} + a_{i(n+i)})^2}{4a_{ii}})x_{n+i}^2] + \ldots +$$
$$[(\sqrt{a_{nn}}x_n + \frac{a_{(2n)n} + a_{n(2n)}}{2\sqrt{a_{nn}}}x_{2n})^2 + (\frac{4a_{nn}a_{(2n)(2n)} - (a_{(2n)n} + a_{n(2n)})^2}{4a_{nn}})x_{2n}^2]$$

. if we have $\begin{matrix} 4a_{ii}a_{(i+n)(i+n)} > (a_{i(i+n)} + a_{(i+n)i})^2 & \text{if i} < \text{n} \\ 4a_{ii}a_{(i-n)(i-n)} > (a_{i(i-n)} + a_{(i-n)i})^2 & \text{if i} \geq \text{n} \end{matrix}$, then the above formula is greater than zero for any X. Therefore, based on the definition of a positive definite matrix, the Matrix A is positive definite. Theorem 1 is proved.

The two classes of users BPR travel cost function used in Mahmassani and Mouskos (1987) (Formula 4.2.2, 4.2.3) fails to satisfy Theorem 1 condition and its Jacobian matrix is not positive definite, as shown below:

Copying the same notation in Formula 4.2.6, 4.2.7, 4.2.8 and 4.2.10, we have the first derivative of the Formula 4.2.2 and 4.2.3 as:

$$\frac{\partial\ t_{ai}}{\partial\ x_{ai}} = \frac{\alpha_{ai}\beta_{ai}t_{ai}^0}{c_i}\left(\frac{x_{ai} + \varepsilon(x_{bi})}{c_i}\right)^{\beta_{ai}-1}$$

$$\frac{\partial\ t_{ai}}{\partial\ x_{bi}} = \frac{\varepsilon\alpha_{ai}\beta_{ai}t_{ai}^0}{c_i}\left(\frac{x_{ai} + \varepsilon(x_{bi})}{c_i}\right)^{\beta_{ai}-1}$$

$$\frac{\partial\ t_{bi}}{\partial\ x_{ai}} = \frac{\alpha_{bi}\beta_{bi}t_{bi}^0}{c_i}\left(\frac{x_{ai} + \varepsilon(x_{bi})}{c_i}\right)^{\beta_{bi}-1}$$

$$\frac{\partial t_{bi}}{\partial x_{bi}} = \frac{\varepsilon\alpha_{bi}\beta_{bi}t_{bi}^{0}}{c_i}\left(\frac{x_{ai}+\varepsilon(x_{bi})}{c_i}\right)^{\beta_{bi}-1} \qquad (4.2.11)$$

Since we can have :

$$\frac{\partial t_{ai}}{\partial x_{ai}} = a_{ii} \ , \ \frac{\partial t_{ai}}{\partial x_{bi}} = a_{i(i+n)} \ , \ \frac{\partial t_{bi}}{\partial x_{ai}} = a_{(i+n)i} \ ..., \ \frac{\partial t_{bi}}{\partial x_{bi}} = a_{(i+n)(i+n)}$$

, then it yields:

$$4a_{ii}a_{(i+n)(+n)} - (a_{i(i+n)} + a_{(i+n)i})^2 = 4 \times \frac{\alpha_{ai}\beta_{ai}t_{ai}^{0}}{c_i}\left(\frac{x_{ai}+\varepsilon(x_{bi})}{c_i}\right)^{\beta_{ai}-1}$$

$$\times \frac{\varepsilon\alpha_{bi}\beta_{bi}t_{bi}^{0}}{c_i}\left(\frac{x_{ai}+\varepsilon(x_{bi})}{c_i}\right)^{\beta_{bi}-1} - \frac{\varepsilon^2\alpha_{ai}^{2}\beta_{ai}^{2}t_{ai}^{0\,2}}{c_i^{2}}\left(\frac{x_{ai}+\varepsilon(x_{bi})}{c_i}\right)^{2\beta_{ai}-2} -$$

$$\frac{\alpha_{bi}^{2}\beta_{bi}^{2}t_{bi}^{0\,2}}{c_i^{2}}\left(\frac{x_{ai}+\varepsilon(x_{bi})}{c_i}\right)^{2\beta_{bi}-2} -$$

$$2 \times \frac{\alpha_{ai}\beta_{ai}t_{ai}^{0}}{c_i}\left(\frac{x_{ai}+\varepsilon(x_{bi})}{c_i}\right)^{\beta_{ai}-1} \times \frac{\varepsilon\alpha_{bi}\beta_{bi}t_{bi}^{0}}{c_i}\left(\frac{x_{ai}+\varepsilon(x_{bi})}{c_i}\right)^{\beta_{bi}-1}$$

$$= -[\frac{\varepsilon\alpha_{ai}\beta_{ai}t_{ai}^{0}}{c_i}\left(\frac{x_{ai}+\varepsilon(x_{bi})}{c_i}\right)^{\beta_{ai}-1} - \frac{\alpha_{bi}\beta_{bi}t_{bi}^{0}}{c_i}\left(\frac{x_{ai}+\varepsilon(x_{bi})}{c_i}\right)^{\beta_{bi}-1}]^2 \leq 0$$

Therefore, the matrix is not positive definite, which implies that a unique solution to the traffic assignment is not guaranteed. In the following, the LTCF function is examined. Substituting the first derivatives of the LTCF travel cost function (Formula 4.2.10) to the Theorem 1 condition,

$4a_{ii}a_{(i+n)(i+n)} - (a_{i(i+n)} + a_{(i+n)i})^2$, and we have:

$$4a_{ii}a_{(i+n)(i+n)} - (a_{i(i+n)} + a_{(i+n)i})^2 =$$

$$3.5\frac{\alpha_{ai}\beta_{ai}t_{ai}^{0}}{c_i}\times\frac{\varepsilon\alpha_{bi}\beta_{bi}t_{bi}^{0}}{c_i}\times\left(\frac{x_{ai}+0.5\varepsilon x_{bi}}{c_i}\right)^{\beta_{ai}-1}\times\left(\frac{0.5x_{ai}+\varepsilon x_{bi}}{c_i}\right)^{\beta_{bi}-1}-$$

$$\frac{0.25\varepsilon^2\alpha_{ai}^2\beta_{ai}^2(t_{ai}^0)^2}{c_i^2}\left(\frac{x_{ai}+0.5\varepsilon x_{bi}}{c_i}\right)^{2\beta_{ai}-2}-\frac{0.25\alpha_{bi}^2\beta_{bi}^2(t_{bi}^0)^2}{c_i^2}\left(\frac{0.5x_{ai}+\varepsilon x_{bi}}{c_i}\right)^{2\beta_{bi}-2}$$

$$=\frac{1}{4}\left[\frac{\alpha_{bi}^2\beta_{bi}^2(t_{bi}^0)^2}{c_i^2}\left(\frac{0.5x_{ai}+\varepsilon x_{bi}}{c_i}\right)^{2\beta_{bi}-2}\right]\{14\times\frac{\varepsilon\alpha_{ai}\beta_{ai}t_{ai}^0}{\alpha_{bi}\beta_{bi}t_{bi}^0}\times\frac{\left(\frac{x_{ai}+0.5\varepsilon x_{bi}}{c_i}\right)^{\beta_{ai}-1}}{\left(\frac{0.5x_{ai}+\varepsilon x_{bi}}{c_i}\right)^{\beta_{bi}-1}}-$$

$$\left(\frac{\varepsilon\alpha_{ai}\beta_{ai}t_{ai}^0}{\alpha_{bi}\beta_{bi}t_{bi}^0}\right)^2\times\left(\frac{\left(\frac{x_{ai}+0.5\varepsilon x_{bi}}{c_i}\right)^{\beta_{ai}-1}}{\left(\frac{0.5x_{ai}+\varepsilon x_{bi}}{c_i}\right)^{\beta_{bi}-1}}\right)^2-1\}$$

(4.2.12)

If Term 4.2.12 is wanted to be greater than 0, it requires that the following inequality formula holds:

$$7-\sqrt{48}<\frac{\varepsilon\alpha_{ai}\beta_{ai}t_{ai}^0}{\alpha_{bi}\beta_{bi}t_{bi}^0}\times\frac{\left(\frac{x_{ai}+0.5\varepsilon x_{bi}}{c_i}\right)^{\beta_{ai}-1}}{\left(\frac{0.5x_{ai}+\varepsilon x_{bi}}{c_i}\right)^{\beta_{bi}-1}}<7+\sqrt{48} \qquad (4.2.13)$$

The inequality condition (4.2.13) can be easily satisfied in almost all scenarios of the specific problem. For the travel cost functions BPR parameters used in this study, the inequality condition (4.2.13) is held, thus the diagonalization algorithm for the two classes of users traffic assignment using LTCF in this study would converge to a unique equilibrium flow. In the following section, a set of numerical experiments is presented to study the convergence characteristics of the diagonalization algorithm based on the link travel cost functions used in this study.

4.3 Numerical Experiment of Diagonalization Algorithm

In the following example, a two-link-two-node network is used to test the validity and effectiveness of the diagonalization algorithm, including the streamlined version of the diagonalization algorithm, as applied to the two classes of users traffic assignment.

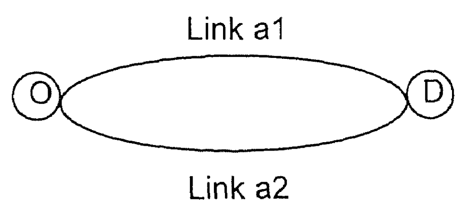


Figure 4.3.1 Example Passenger Car Network

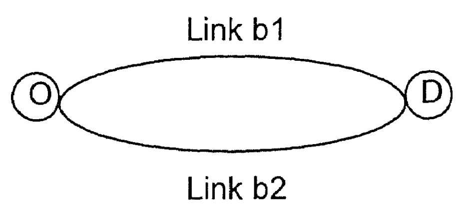


Figure 4.3.2 Example Truck Network

The link performance functions are assumed as:

$$t_{a1} = 2(1+1.03(\frac{x_{a1}+2x_{b1}}{1600})^{5.52})$$

$$t_{a2} = 2(1+1.03(\frac{x_{a2}+2x_{b2}}{2400})^{5.52})$$

$$t_{b1} = 3(1+0.62(\frac{0.5x_{a1}+4x_{b1}}{1600})^{5.14})$$

$$t_{b2} = 3(1+0.62(\frac{0.5x_{a2}+4x_{b2}}{2400})^{5.14})$$

The total O-D flow is 3000 for passenger cars and 300 for trucks, which is:

$q_A = x_{a1} + x_{a2} = 3000$ (passenger cars)

$q_B = x_{b1} + x_{b2} = 300$ (trucks).

Diagonalization Algorithm

To apply the algorithm, an initial feasible solution is needed. Assume that the initial solution is:

$x_{a1}=0$, $x_{a2}=3000$, $x_{b1}=0$, $x_{b2}=300$.

The algorithm's iteration is described as follows:

First Iteration:

Step 1: Diagonalization. Solve the sub problem:

minimize:

$$z(x_{a1},x_{a2},x_{b1},x_{b2}) = \int_0^{x_{a1}}(2+4.24\times10^{-18}(\omega)^{5.52})d\omega + \int_0^{x_{a2}}(2+4.52\times10^{-19}(\omega+2\times300)^{5.52})d\omega + \int_0^{x_{b1}}(3+6.31\times10^{-17}(4\omega)^{5.14})d\omega + \int_0^{x_{b2}}(2+7.86\times10^{-18}(0.5\times3000+4\omega)^{5.14})d\omega$$

Subject to: xa1+xa2=3000

xb1+xb2=300

xa1, xa2, xb1, xb2 >=0.

The Frank Wolfe algorithm is used to solve the problem, and the solution of step 1 is:

x_{a1}=1440.001, x_{a2}=1559.999, x_{b1}=270.0027, x_{b2}=29.9973

Step 2 : Convergence Test.

$$\max_a \left\{ \frac{x_a^1 - x_a^0}{x_a^0} \right\} = \frac{x_{a1}^1 - x_{a1}^0}{x_{a1}^0} = \infty$$

A convergence criteria of 1% is used, so the first iteration does not meet the criteria.

Second Iteration:

Step 1: Diagonalization. Solve the sub-problem

minimize:

$$z(x_{a1}, x_{a2}, x_{b1}, x_{b2}) = \int_0^{x_{a1}} (2 + 4.24 \times 10^{-18} (\omega + 2 \times 270.0027)^{5.52}) d\omega +$$

$$\int_0^{x_{a2}} (2 + 4.52 \times 10^{-19} (\omega + 2 \times 29.9973)^{5.52}) d\omega +$$

$$\int_0^{x_{b1}} (3 + 6.31 \times 10^{-17} (4\omega + 0.5 \times 1440.001)^{5.14}) d\omega +$$

$$\int_0^{x_{b2}} (3 + 7.86 \times 10^{-18} (0.5 \times 1559.999 + 4\omega)^{5.14}) d\omega$$

Subject to: xa1+xa2=3000

xb1+xb2=300

xa1, xa2, xb1, xb2 >=0.

The solution is: xa1=899.9986, xa2=2100.0014,xb1=89.99863, xb2=210.0014.

Step 2 Convergence Test.

$$\max_a \left\{ \frac{x_a^2 - x_a^1}{x_a^1} \right\} = \frac{x_{b2}^2 - x_{b2}^1}{x_{b2}^1} = \frac{210.0014 - 29.9973}{29.9973} = 6.00688$$

The convergence criteria is not met.

A summary of 11 algorithmic iterations is shown in Table 4.3.1. This table displays the iteration number, the solution of the sub-problem, and two convergence measures, one of which is the convergence to the final solution.

Table 4.3.1 shows that the algorithm converges to the correct equilibrium solution. The convergence measure with respect to the previous solution shows a monotonous

Table 4.3.1 Iteration of the Diagonalization Procedure for the Example

Iteration	x_{1a}^n	x_{2a}^n	x_{1b}^n	x_{2b}^n	$\max\left\{\frac{x_a^n - x_a^{n-1}}{x_a^{n-1}}\right\}$	$\max\left\{\frac{x_a^n - x^*}{x^*}\right\}$
0	0	3000	0	300	-	1
1	1440	1600	270	30	∞	0.2
2	900	2100	90	210	6	0.25
3	1260	1740	158	142	0.7556	0.05
4	1125	1875	113	187	0.3169	0.0625
5	1215	1785	130	170	0.1504	0.0125
6	1181	1819	118	182	0.0923	0.0158
7	1204	1796	122	178	0.0339	0.0033
8	1196	1804	120	180	0.0164	0.0033
9	1201	1799	121	179	0.0083	0.0008
10	1199	1801	120	180	0.0083	0.0008
11	1200	1800	120	180	0.0008	0.00

and asymptotic convergence rate, while the actual converge measure with respect to the final solution) rate is not monotonous, but it finally converges to zero.

In this example, solving the sub-problem uses only two to three iterations before the sub-problem converges. Therefore, a modified version of the diagonalization algorithm, the streamlined algorithm which just uses one iteration for every sub-problem, is suggested to reduce the number of iterations. Following, the streamlined algorithm is presented to solve the example problem.

Streamlined Algorithm

The initial solution is: $x_{1a}^0 = 0$, $x_{2a}^0 = 3000$,, $x_{1b}^0 = 0$, and $x_{2b}^0 = 300$.

First Iteration:

Step1: $t_{1a}^0 = 2$, $t_{2a}^0 = 21.315$, $t_{1b}^0 = 3$, $t_{2b}^0 = 6.408$

Step 2 : Since $t_{1a}^0 < t_{2a}^0$ and $t_{1b}^0 < t_{2b}^0$, we have $y_{1a}^0 = 3000$, $y_{2a}^0 = 0$, $y_{1b}^0 = 300$, and $y_{2b}^0 = 0$.

Step3:

$$\min_{0 \le \alpha a, \alpha b \le 1} \int_0^{3000\alpha a} (2 + 4.24 \times 10^{-18} (\omega)^{5.52}) d\omega + \int_0^{3000-3000\alpha a} (2 + 4.52 \times 10^{-19} (\omega + 2 \times 300)^{5.52}) d\omega +$$
$$\int_0^{300\alpha b} (3 + 6.31 \times 10^{-17} (4\omega)^{5.14}) d\omega + \int_0^{300-300\alpha b} (2 + 7.86 \times 10^{-18} (0.5 \times 3000 + 4\omega)^{5.14}) d\omega$$

The solution is $\alpha_a = 0.48$, and $\alpha_b = 0.9$.

Step 4: $x_{1a}^1 = 1440.001$, $x_{2a}^1 = 1559.999$,, $x_{1b}^1 = 270.003$, and $x_{2b}^1 = 29.997$.

Step 5: Go to Step 1.

A summary of 10 algorithm iterations is shown in Table 4.3.2. In comparison to the solution obtained by the original diagonalization algorithm (Table 4.3.1), it may be observed in this specific example, the solutions of the two algorithms in each iteration is very close and both of them approach to the final equilibrium flow. For the example shown in Sheffi (1985), though the two algorithms' each iteration solution is

Table 4.3.2 Iterations of the Streamlined Algorithm Procedure for the Example

Iteration	x_{1a}^n	x_{2a}^n	x_{1b}^n	x_{2b}^n	$\max\left\{\frac{x_a^n - x_a^{n-1}}{x_a^{n-1}}\right\}$	t_{1a}^n	t_{2a}^n	t_{1b}^n	t_{2b}^n
0	0	3000	0	300	-	2	21.315	3	6.408
1	1440.001	1559.999	270.003	29.997	∞	8.679	2.235	6.408	3.012
2	899.999	2100.001	89.999	210.001	6	2.235	4.697	3.056	3.545
3	1259.998	1740.002	157.5	142.5	0.75	3.895	2.804	3.550	3.134
4	1124.001	1875.999	112.5	187.5	0.3158	2.803	3.446	3.177	3.305
5	1214.999	1785.001	129.499	170.501	0.1511	3.310	3.055	3.282	3.225
6	1180.999	1819.001	118.125	181.875	0.0878	3.055	3.220	3.240	3.251
7	1203.751	1796.249	122.375	177.625	0.0360	3.190	3.127	3.260	3.234
8	1195.25	1804.75	119.531	180.469	0.0232	3.127	3.168	3.242	3.250
9	1200.938	1799.062	120.594	179.406	0.0089	3.161	3.145	3.249	3.245
10	1198.813	1801.187	119.883	180.117	0.0059	3.145	3.156	3.245	3.248

Iteration	y_{1a}^n	y_{2a}^n	y_{1b}^n	y_{2b}^n	α_a	α_b	$\max\left\{\frac{x_a^n - x_a^*}{x_a^*}\right\}$
0	3000	0	300	0	0.48	0.9	1
1	0	3000	0	300	0.375	0.6667	1.25
2	3000	0	300	0	0.1714	0.3214	0.25
3	0	3000	0	300	0.1080	0.2880	0.3125
4	3000	0	300	0	0.0485	0.0907	0.0637
5	0	3000	0	300	0.0280	0.0878	0.0792
6	3000	0	300	0	0.0125	0.0234	0.0158
7	0	3000	0	300	0.0071	0.0232	0.0198
8	3000	0	300	0	0.0032	0.0059	0.0040
9	0	3000	0	300	0.0018	0.0059	0.0050
10	3000	0	300	0	0.0008	0.0015	0.001

Table 4.3.3 Iterations of the Streamlined Algorithm for the Example with Different Initial Solution (1)

Iteration	x^n_{1a}	x^n_{2a}	x^n_{1b}	x^n_{2b}	$\max\left\{\frac{x^n_a - x^{n-1}_a}{x^{n-1}_a}\right\}$	t^n_{1a}	t^n_{2a}	t^n_{1b}	t^n_{2b}
0	3000	0	300	0	-	183.098	2	30.387	3
1	839.999	2160.001	0.0046	299.995	∞	2.059	6.456	3.002	4.429
2	1439.991	1560.009	165.003	134.997	35869	5.597	2.461	3.870	3.086
3	1109.995	1890.005	90.002	209.998	0.5556	2.627	3.668	3.105	3.406
4	1259.997	1740.003	131.250	168.750	0.4583	3.566	2.929	3.348	3.193
5	1177.499	1822.501	112.501	187.499	0.1429	2.995	3.263	3.202	3.281
6	1214.998	1785.002	122.813	177.187	0.0912	3.246	3.093	3.269	3.232
7	1194.374	1805.626	118.126	181.8743	0.0381	3.111	3.179	3.235	3.255
8	1203.75	1796.25	120.703	179.297	0.0218	3.175	3.136	3.252	3.243
9	1198.594	1801.406	119.532	180.468	0.0097	3.141	3.158	3.243	3.249
10	1200.936	1799.064	120.176	179.824	0.0054	3.157	3.148	3.248	3.246

Iteration	y^n_{1a}	y^n_{2a}	y^n_{1b}	y^n_{2b}	α_a	α_b	$\max\left\{\frac{x^n_a - x^*_a}{x^*_a}\right\}$
0	0	3000	0	300	0.72	0.999	1.5
1	3000	0	300	0	0.2778	0.5500	0.9999
2	0	3000	0	300	0.2292	0.4545	0.3750
3	3000	0	300	0	0.0794	0.1964	0.2500
4	0	3000	0	300	0.0655	0.1429	0.0938
5	3000	0	300	0	0.0206	0.0549	0.0625
6	0	3000	0	300	0.0170	0.0382	0.0234
7	3000	0	300	0	0.0052	0.0142	0.0156
8	0	3000	0	300	0.0043	0.0097	0.0059
9	3000	0	300	0	0.0013	0.0036	0.0039
10	0	3000	0	300	0.0011	0.0024	0.0015

Table 4.3.4 Iterations of the Streamlined Algorithm for the Example with Different Initial Solution (2)

Iteration	x_{1a}^n	x_{2a}^n	x_{1b}^n	x_{2b}^n	$\max\left\{\frac{x_a^n - x_a^{n-1}}{x_a^{n-1}}\right\}$	t_{1a}^n	t_{2a}^n	t_{1b}^n	t_{2b}^n
0	1500	1500	150	150	-	5.947	2.421	3.777	3
1	1139.999	1860.001	82.500	217.500	0.45	2.669	3.609	3.097	3.424
2	1275	1725	127.500	172.500	0.5455	3.609	2.911	3.337	3.198
3	1185	1815	110.627	189.374	0.1323	3.010	3.254	3.198	3.284
4	1218.748	1781.252	121.875	178.125	0.1017	3.254	3.087	3.267	3.234
5	1196.25	1803.75	117.657	182.343	0.0350	3.115	3.177	3.234	3.256
6	1204.688	1795.312	120.469	179.531	0.0239	3.177	3.135	3.252	3.243
7	1199.062	1800.938	119.414	180.586	0.0088	3.142	3.158	3.243	3.249
8	1201.173	1798.827	120.117	179.883	0.0059	3.158	3.147	3.248	3.246
9	1199.765	1800.235	119.854	180.146	0.0022	3.149	3.153	3.246	3.247
10	1200.292	1799.708	120.029	179.971	0.0015	3.153	3.151	3.247	3.246

Iteration	y_{1a}^n	y_{2a}^n	y_{1b}^n	y_{2b}^n	α_a	α_b	$\max\left\{\frac{x_a^n - x_a^*}{x_a^*}\right\}$
0	0	3000	0	300	0.2400	0.4500	0.25
1	3000	0	300	0	0.0726	0.2069	0.3125
2	0	3000	0	300	0.0706	0.1323	0.0625
3	3000	0	300	0	0.0186	0.0594	0.0781
4	0	3000	0	300	0.0185	0.0346	0.0156
5	3000	0	300	0	0.0047	0.0154	0.0195
6	0	3000	0	300	0.0047	0.0088	0.0039
7	3000	0	300	0	0.0012	0.0039	0.0049
8	0	3000	0	300	0.0012	0.0022	0.0010
9	3000	0	300	0	0.0003	0.0010	0.0012
10	0	3000	0	300	0.0003	0.0005	0.0002

not as close, the streamlined algorithm almost performs as well as the diagonalization algorithm. Table 4.3.3 and Table 4.3.4 are the summary of the streamlined diagonalization algorithm on the above example using different initial solution states is presented. Contrary to the initial stating solution used above, one starting solution has all the traffic load in Link 1a and Link 1b. The other starting solution has the traffic load equally distributed over two passenger car links and trucks.

The results from Table 4.3.3-Table 4.3.4 show that despite utilizing different solution states, the streamlined algorithm leads the solutions to the equilibrium flow with a very similar convergence rate. Furthermore, as it was proved earlier that the link performance function (Formula 4.2.4 and 4.2.5) Jacobian matrix is positive definite, which implies that the UE solution found is also a unique solution. It may be concluded that the link performance function (Formula 4.2.4 and 4.2.5) used in this study would allow the diagonalization algorithm to converge to the problem unique equilibrium solution.

4.4 Numerical Experiment of Diagonalization Algorithm on Five Test Networks

In this section, the performance of the diagonalization algorithm for two classes of users traffic assignment is tested on five networks.

4.4.1 Five Test Networks

Five different size networks, ranging from 16 links to 110 links for single class network (Figure 4.4.1-4.4.5), are used as the test networks of this study.

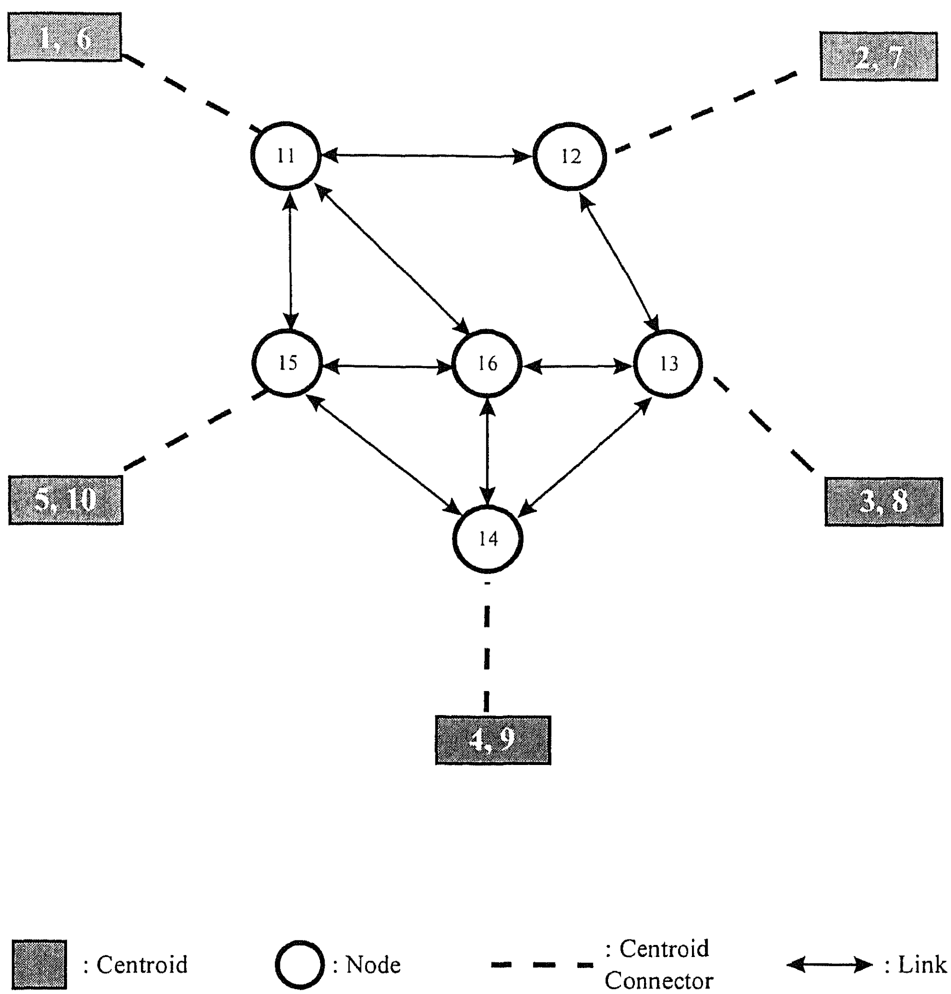


Figure 4.4.1 Network 1

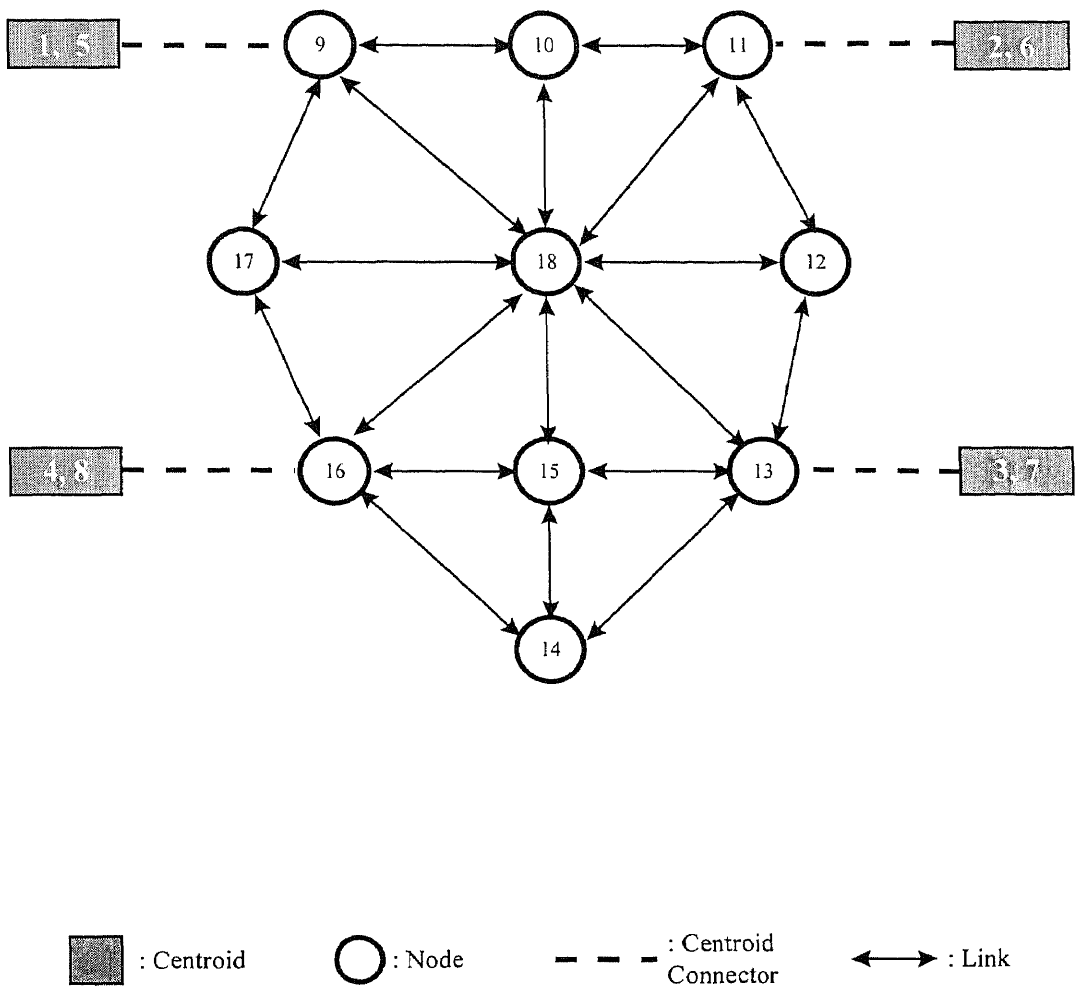


Figure 4.4.2 Network 2

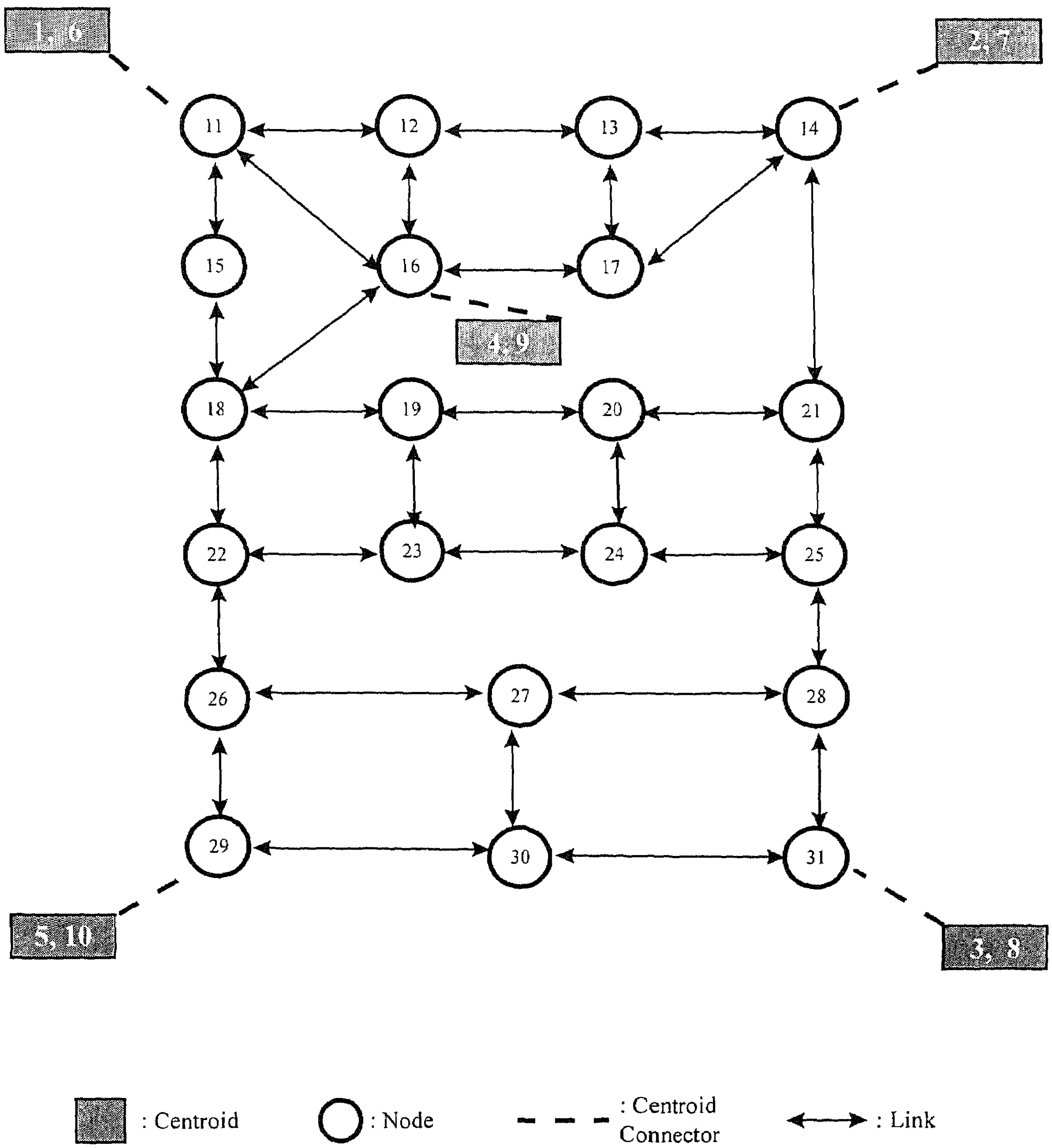


Figure 4.4.3 Network 3

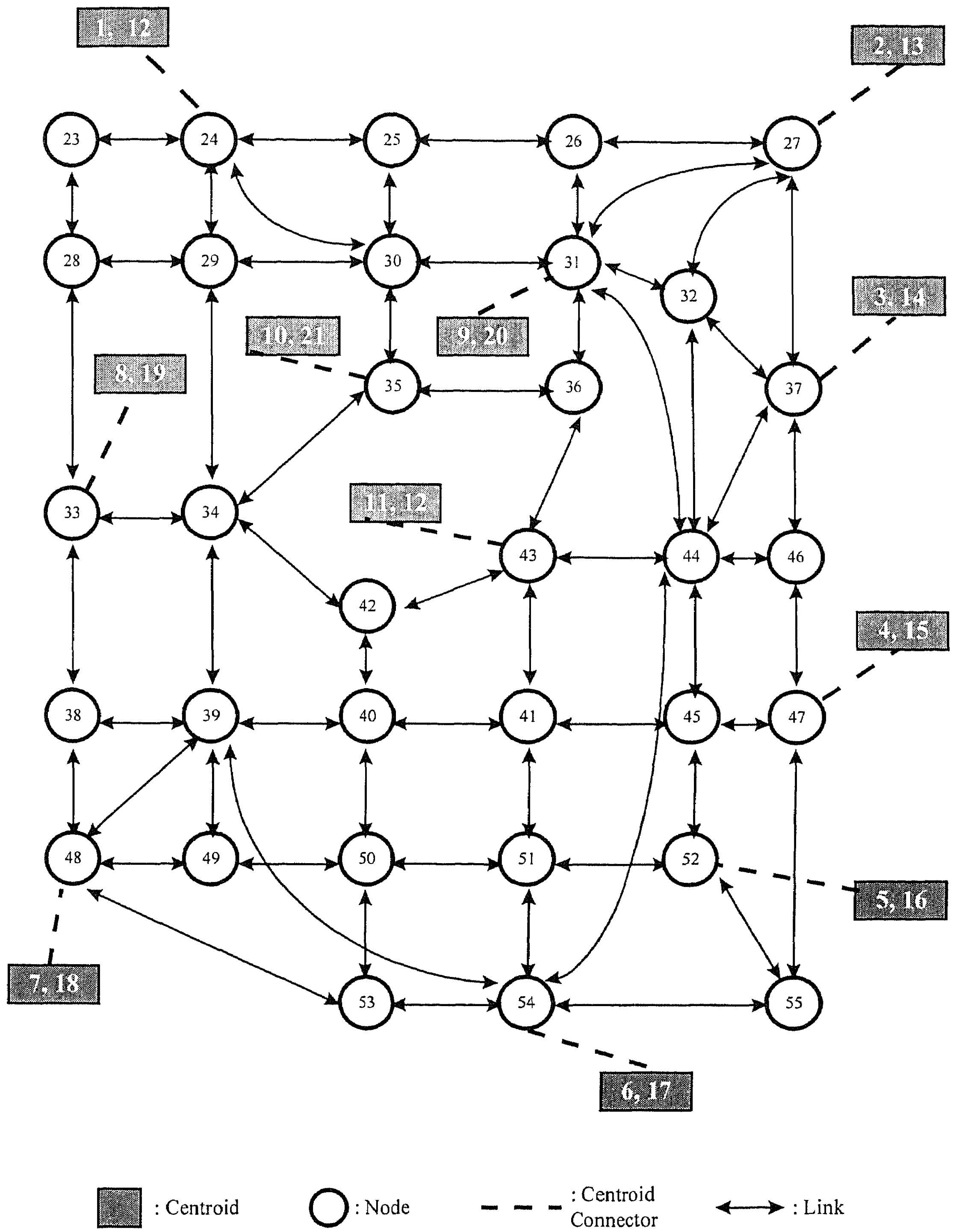


Figure 4.4.4 Network 4

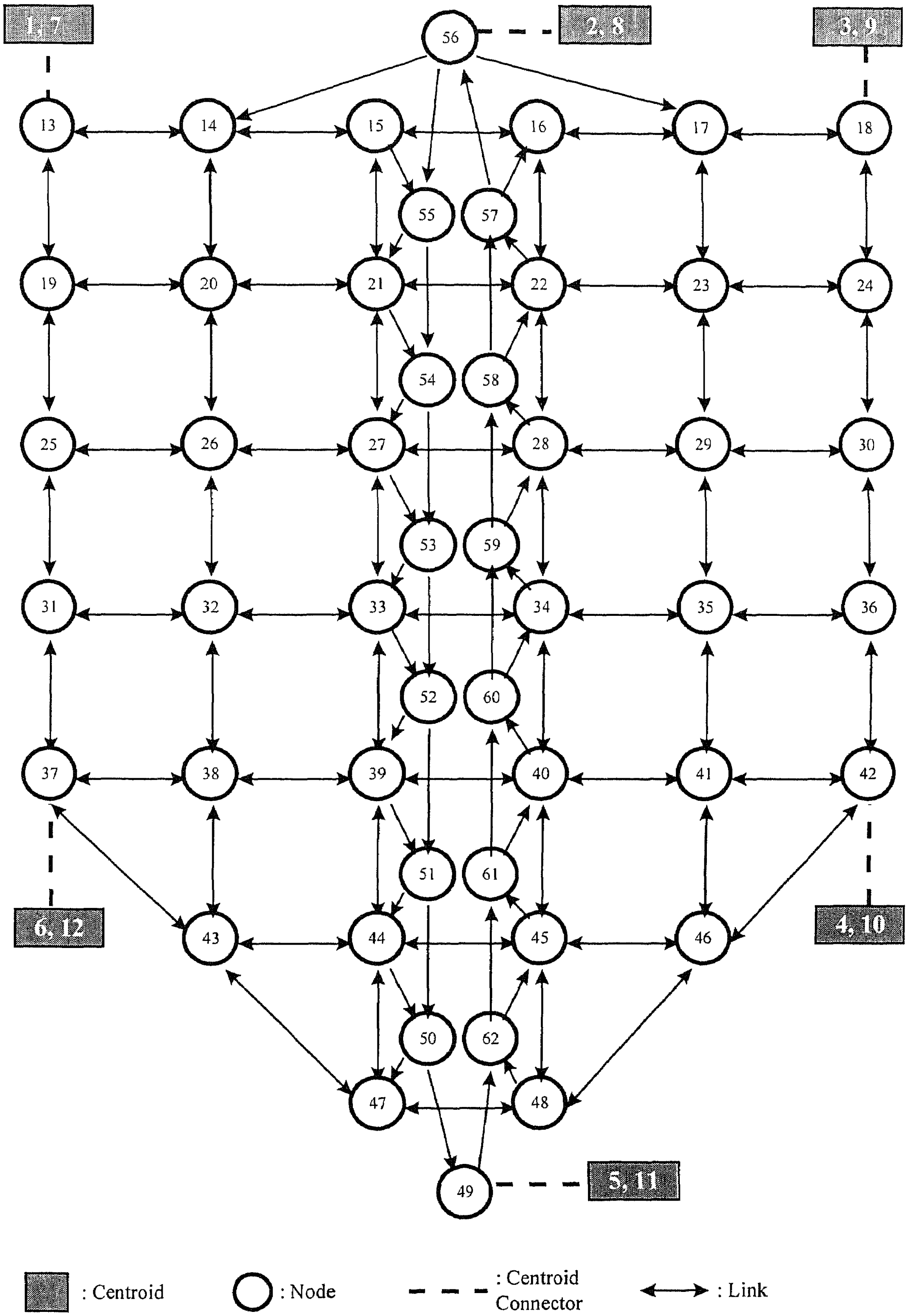


Figure 4.4.5 Network 5

The network used have similar settings as the common transportation networks. As shown in Figures 4.4.1-4.4.5, the centroids are the network users origins and destinations and the nodes are the intersections of the physical roadway. The centroid is connected to the network by the centroid connectors, which have infinite capacities and no travel time costs. Most of the links represent single direction traffic movements. The database for the attributes of each link include the links length (also shown in the graphs), link capacity, type of links (determine the free flow travel speed and the coefficients used in the travel time cost function) and the capacity for the additional lane. The two classes of users networks contains two layers of the networks, one of which is for the passenger cars and the other is for the trucks.

Network 1 is illustrated in Figure 4.4.1. It has 6 nodes, 18 links and 5 centroids for single class network and 12 nodes, 36 links with 10 centroids for the two classes network.

Network 2 contains 10 nodes and 38 links , as well as 4 centroids for single class network and double number of nodes, links and centroids for two classes network. Network 2 is more than two times the size of the Network 1.

Network 3 is more complicated than Network 1 and 2, which has 21 nodes, 64 links and 5 centroids and double number of two classes of users. It is constructed in the way to simulate the typical urban transportation journey to work trip structure that is from radial spreading suburban residential areas to a central business district.

Network 4 is doubles the size of Network 3 and has a similar structure. It has 33 nodes, 126 links and 11 centroids for the single class network. Likewise, the two classes network has double numbers of nodes, links and centroids.

Network 5 is formed by 50 nodes, 166 links and 6 centroids for single class network and 100 nodes, 332 links and 12 centroids for two classes network. Network 5 has a freeway network structure, a primary freeway and parallel service roads.

4.4.2 Test of Convergence and Convergence Rate with Number of Internal Iteration

One of the important issues arising from Section 4.3 is that since the degree of the convergence of the sub problem does not have a significant impact on the diagonalization algorithm's final convergence to the solution, what is the best convergence degree such that the algorithm can most efficiently converge to its final solution. In the computer programmed diagonalization algorithm, the sub problem convergence degree is interpreted as the number of internal iterations.

In this numerical experiment, we apply the computer programmed diagonalization algorithm on the five test networks (Figures 4.4.1-4.4.5) with different number of internal iterations. The criteria for terminating the program procedure is when the final convergence reaches 0.001, that is the ratio of total link flow change during the last two iterations against the current link flow. The result is shown in Figure 4.4.6-4.4.25.

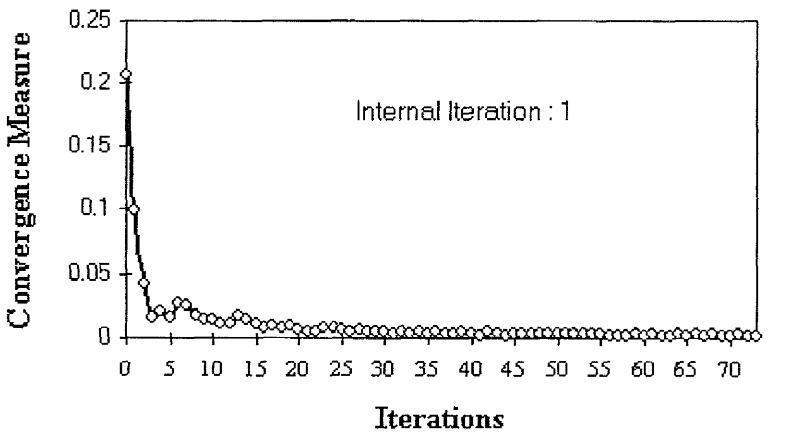


Figure 4.4.6 Network 1 Convergence vs Iter. (1 Internal Iter.)

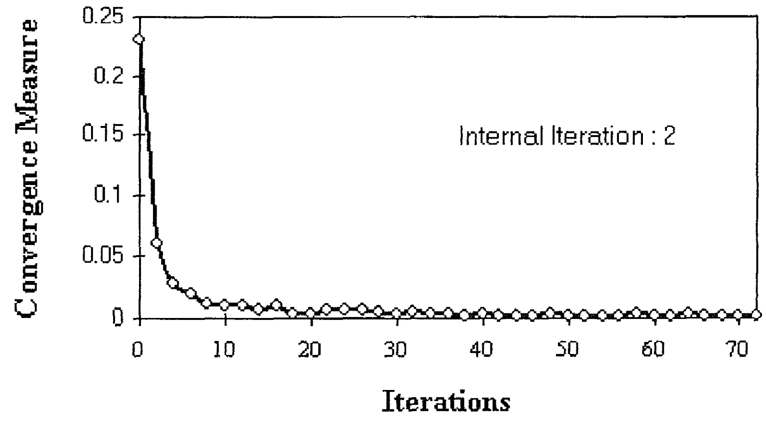


Figure 4.4.7 Network 1 Convergence vs Iter. (2 Internal Iter.)

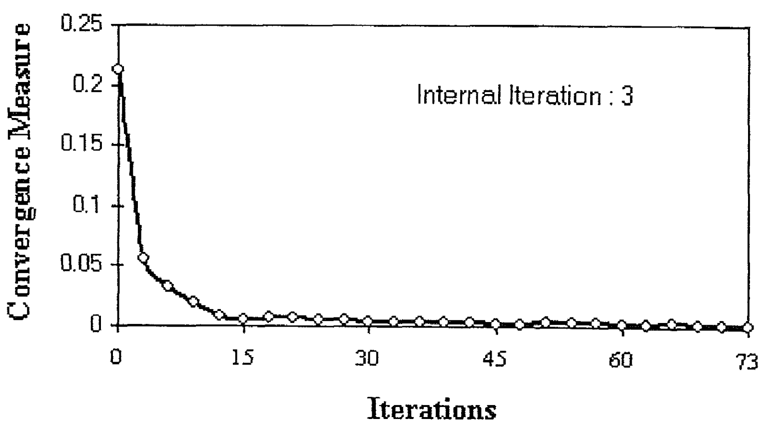


Figure 2.4.8 Network 1 Convergence vs Iter. (3 Internal Iter.)

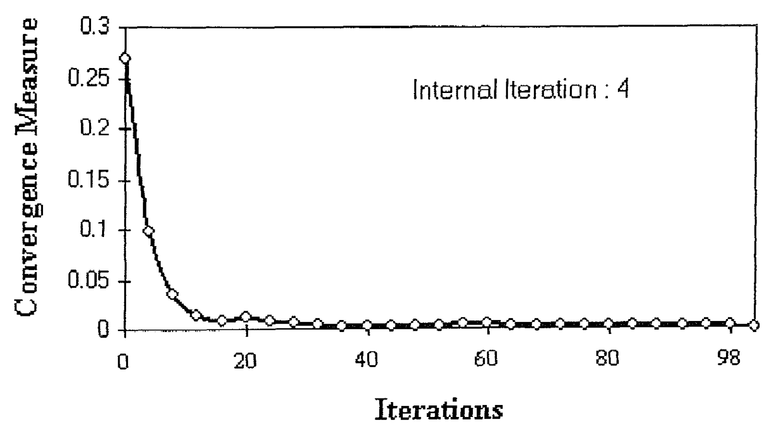


Figure 4.4.9 Network 1 Convergence vs Iter. (4 Internal Iter.)

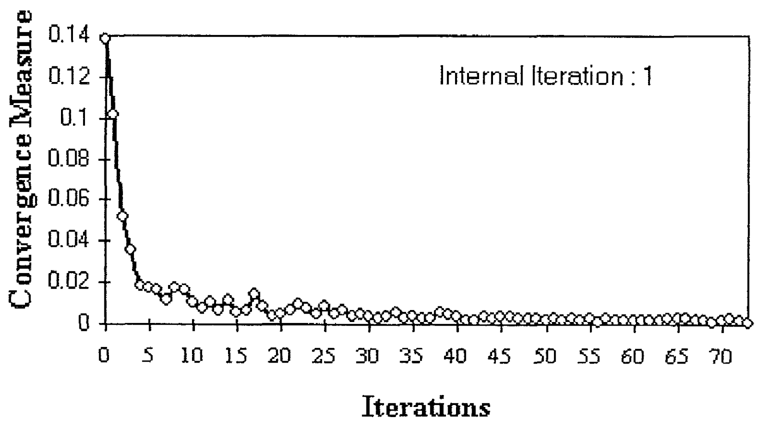


Figure 4.4.10 Network 2 Convergence vs Iter. (1 Internal Iter.)

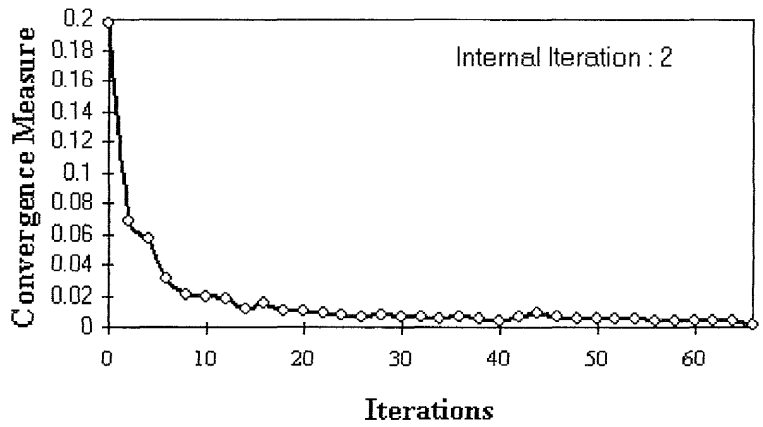


Figure 4.4.11 Network 2 Convergence vs Iter. (2 Internal Iter.)

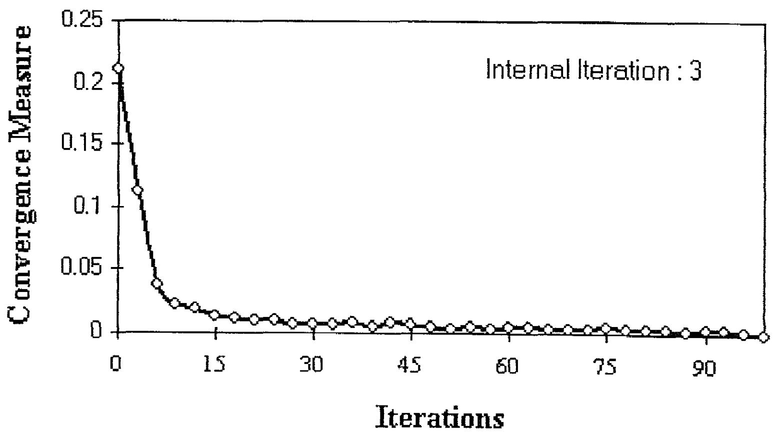


Figure 4.4.12 Network 2 Convergence vs Iter. (3 Internal Iter.)

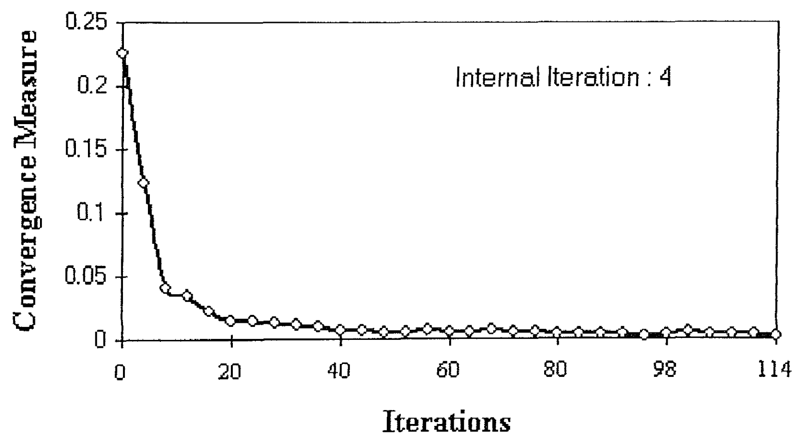


Figure 4.4.13 Network 2 Convergence vs Iter. (4 Internal Iter.)

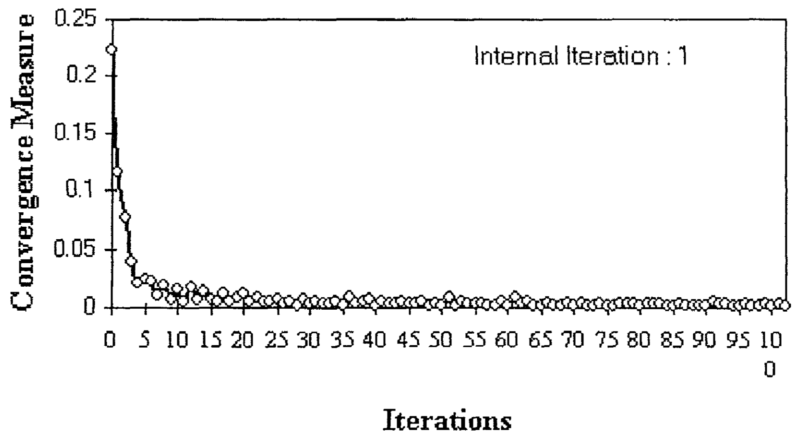


Figure 4.4.14 Network 3 Convergence vs Iter. (1 Internal Iter.)

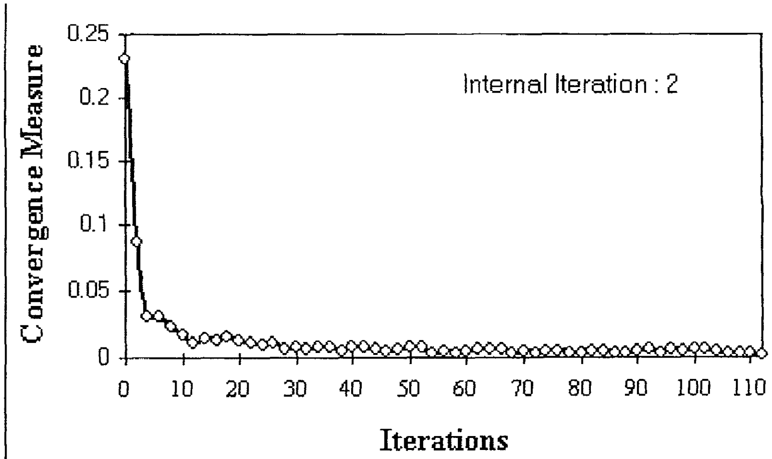


Figure 4.4.15 Network 3 Convergence vs Iter. (2 Internal Iter.)

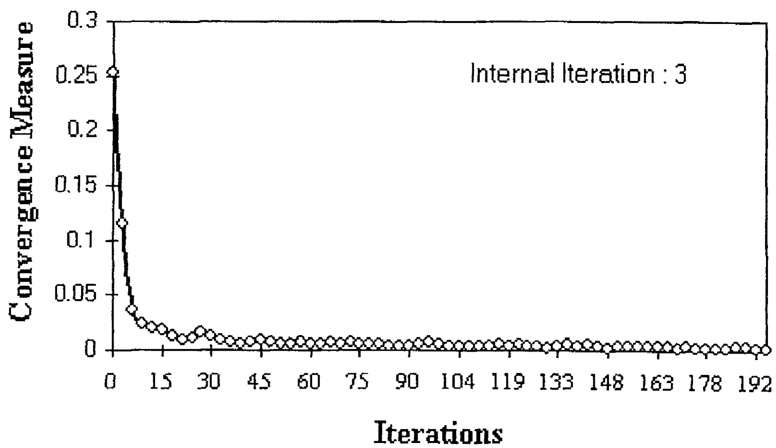


Figure 4.4.16 Network 3 Convergence vs Iter. (3 Internal Iter.)

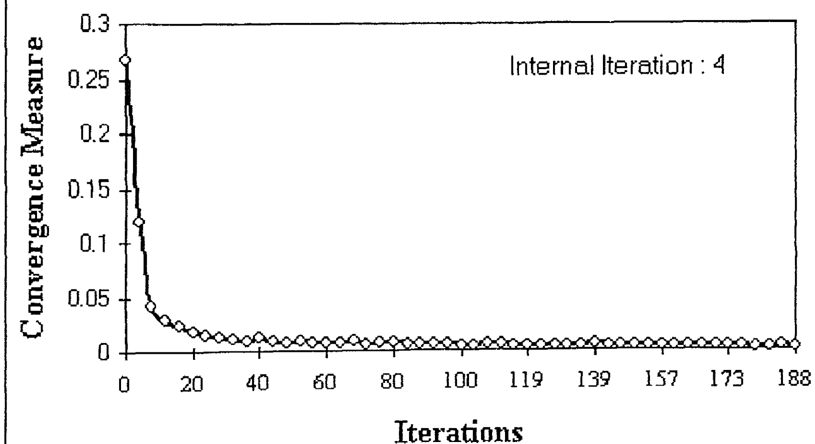


Figure 4.4.17 Network 3 Convergence vs Iter. (4 Internal Iter.)

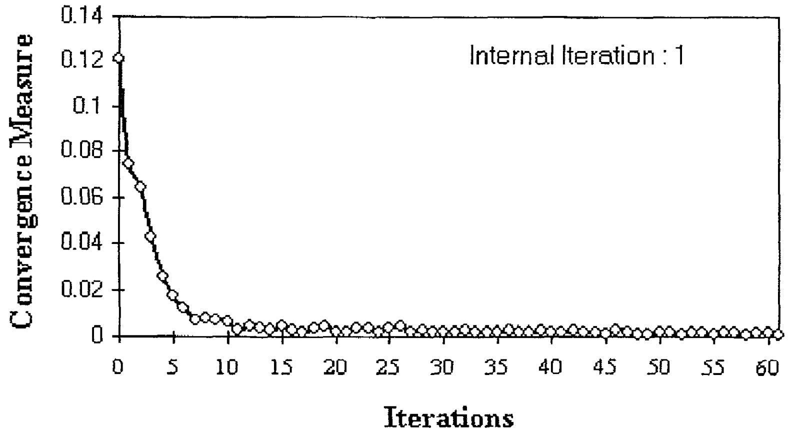


Figure 4.4.18 Network 4 Convergence vs Iter. (1 Internal Iter.)

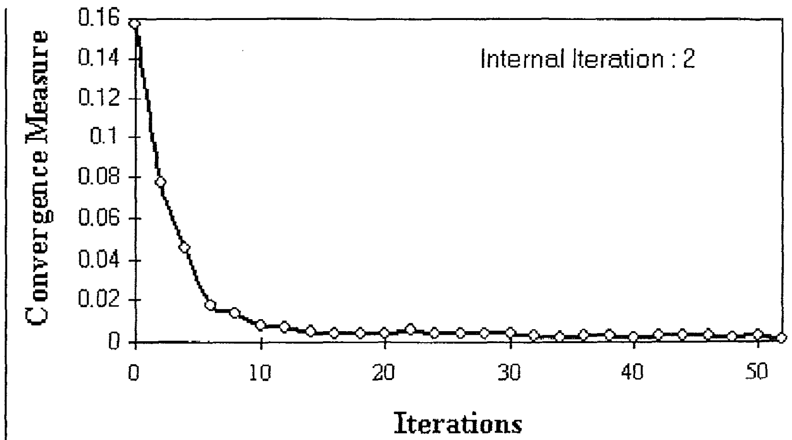


Figure 4.4.19 Network 4 Convergence vs Iter. (2 Internal Iter.)

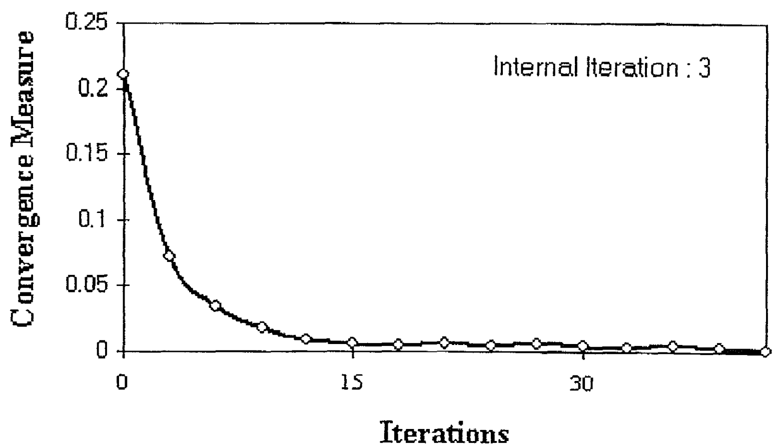


Figure 4.4.20 Network 4 Convergence vs Iter. (3 Internal Iter.)

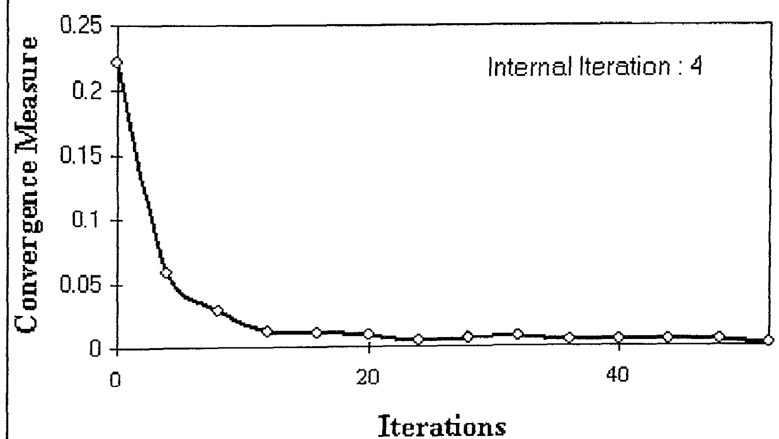


Figure 4.4.21 Network 4 Convergence vs Iter. (4 Internal Iter.)

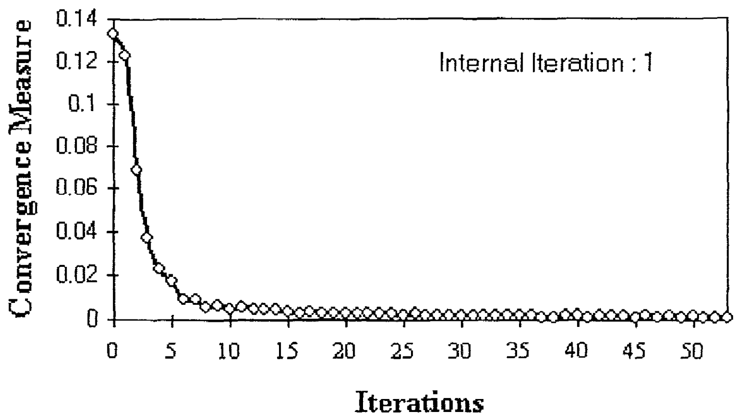


Figure 4.4.22 Network 5 Convergence vs Iter. (1 Internal Iter.)

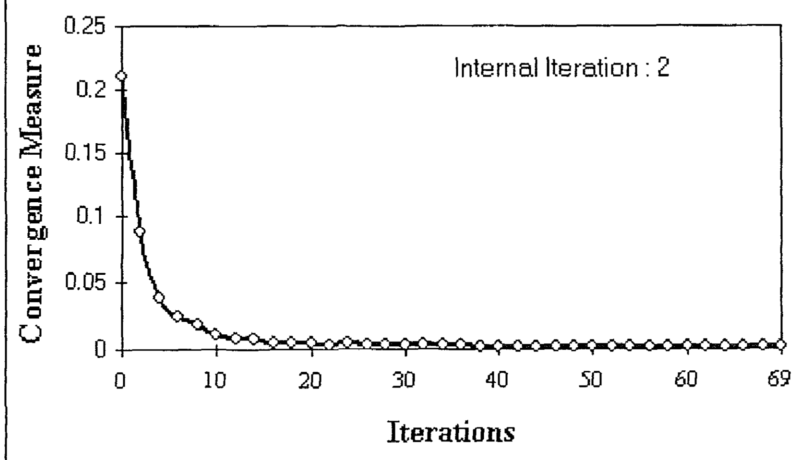


Figure 4.4.23 Network 5 Convergence vs Iter. (2 Internal Iter.)

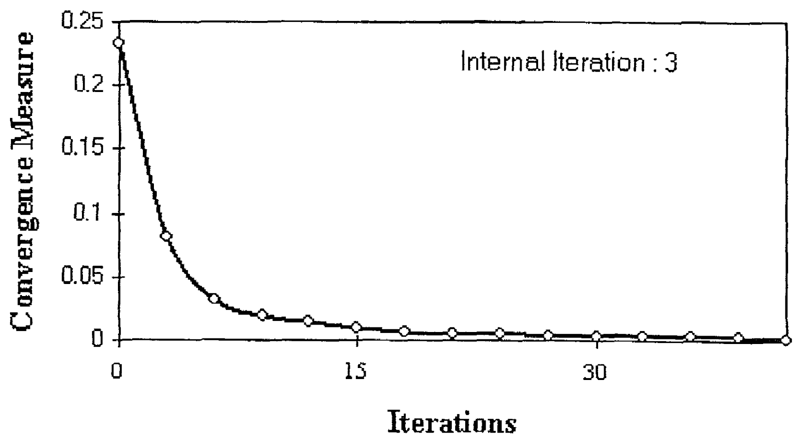


Figure 4.4.24 Network 5 Convergence vs Iter. (3 Internal Iter.)

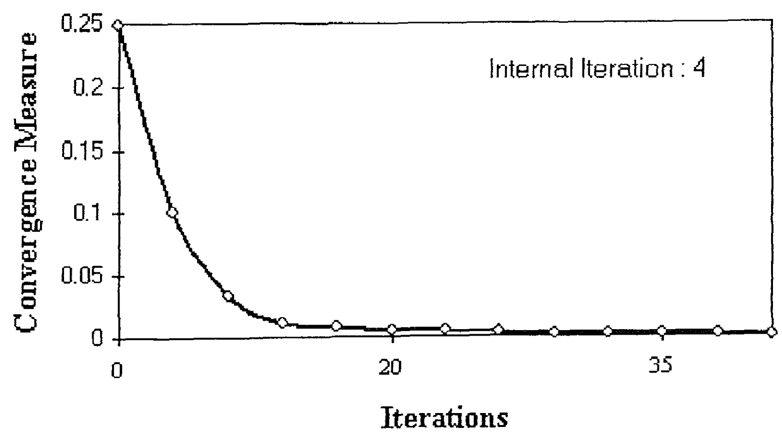


Figure 4.4.25 Network 5 Convergence vs Iter. (4 Internal Iter.)

From this result, the following observations are made:

1) The diagonalization algorithm does converge in all five networks with different internal iteration number, though none of them demonstrate any monotone convergence property. This observation is consistent with the findings in Section 4.3, where the manual calculation approach is applied.

2) The algorithm fast approaches to certain point during the first few iterations, and then it slowly and gradually reaches the final convergence criteria. In Network 1 experiments, the algorithm quickly converge from 0.2 to the neighborhood of 0.01 within first 15 iterations, and then use over 70 iterations to reach 0.001 (See Figures 4.4.6-4.4.9). Networks 2, 3, 4 and 5 have very similar convergence pattern (See Figures 4.4.10-4.4.25). Thus, different convergence measurements established by the specific problems would be a dominant factor in determining the speed of the diagonalization algorithm.

3) Network configuration does not seem to affect the general properties of the diagonalization algorithm application. The five networks have very different network sizes and general network structures, but all the graphs of the experiments have very similar pattern as the computing iteration process approaches to the termination. This confirms the results of Mahmassani and Mouskos (1986), even though the Jacobian matrix in their application is not positive definite.

4) There does not exists a best internal iteration number. In Network 1, the algorithm converged to 0.001 at around 70 iterations when the internal iteration numbers are

1, 2 and 3, while 4 internal iteration needs over 100 overall iterations to reach convergence 0.001. Internal iteration equal to 2 though is the best one for Network 2. In Network 3, the algorithm with internal iteration equal to 1 or 2 is twice efficient than internal iteration equal to 3 or 4. Conversely, in Network 4 and Network 5, the greater the internal iteration number is, the faster the algorithm proceeds. These observations implicitly suggest that the best internal iteration number is more or less determined by the problem itself. With complicated network structure, it requires more internal iteration for the sub-problem solution, and the small number iteration would cause the solution approach the optimal solution with high degree of fluctuation, requiring more iterations to converge.

Observation 4 findings conflict with Sheffi 's proposal that the streamline algorithm which only uses one internal iteration, is a more efficient algorithm by reducing the iteration number, which is also reported in Mahmassani and Mouskos (1986). In this study, the computerized diagonalization algorithm with internal iteration between 2 and 4 is used, which corresponds to the findings of the numerical experiments.

4.4.3 Test of Network Flow Equilibrium and Distribution Pattern

In this series of numerical experiments, the goal is to find whether the diagonalization algorithm converges to the user equilibrium flow and study the flow distribution. Both passenger car trips and truck trips are examined on the five test networks, and the results are summarized in Table 4.4.1 - Table 4.4.7.

Table 4.4.1 Network 1: Passenger Car Path for Car Only Trips

PATH 1			PATH 2		
NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW
11			11		
15	0.0323	555	16	0.0333	445
14	0.0567	555	14	0.0568	445

Table 4.4.2 Network 1: Truck Path for Truck Only Trips

PATH 1			PATH 2		
NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW
11			11		
15	0.0323	145	16	0.0333	105
14	0.0567	145	14	0.0567	105

Table 4.1.3 Network 1: Passenger Car Path for Combined Trips

PATH 1			PATH 2		
NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW
11			11		
15	0.0436	599	16	0.0333	401
14	0.0681	599	14	0.0679	401

Table 4.4.4 Network 1: Truck Path for Combined Trips

PATH 1			PATH 2		
NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW
11			11		
15	0.0495	159	16	0.0333	91
14	0.074	159	14	0.0751	91

Table 4.4.5 Network 3: Passenger Car Path for Car Only Trips

PATH 1		
NODES	CULM. TIME (Minutes)	FLOW
12		
13	0.04	1,000
14	0.08	1,000
21	0.1601	1,000
25	0.2001	1,000
28	0.3055	1,000
31	0.3389	1,000

Table 4.4.6 Network 3: Truck Path for Truck Only Trips

PATH 1			PATH 2			PATH 3		
NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW
14			14			14		
21	0.0975	595	17	0.0596	405	17	0.0596	405
25	0.1462	595	16	0.1269	405	16	0.1269	405
28	0.4372	595	18	0.1865	405	18	0.1865	405
31	0.4722	595	22	0.2272	405	22	0.2272	405
			26	0.2637	405	26	0.2637	405
			29	0.2819	60	27	0.3803	344
			30	0.3819	60	30	0.3824	344
			31	0.4728	405	31	0.4733	405

Table 4.4.7 Network 3: Passenger Car Path for Combined Trips

PATH 1			PATH 2			PATH 3		
NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW
12			12			12		
13	0.04	737	11	0.04	263	11	0.04	263
14	0.08	737	15	0.08	263	15	0.08	263
21	0.1722	737	18	0.12	263	18	0.12	263
25	0.2233	737	22	0.1601	263	22	0.1601	263
28	0.3753	737	26	0.1965	263	26	0.1965	263
31	0.4165	737	27	0.2974	130	29	0.2147	134
			30	0.3156	130	30	0.3157	134
			31	0.4162	263	31	0.4163	263

Table 4.4.8 Network 3: Truck Path for Combined Trips

PATH 1			PATH 2			PATH 3			PATH 4			PATH 5		
NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW
14			14			14			14			14		
21	0.1031	529	17	0.0613	434	17	0.0613	434	13	0.037	37	13	0.037	37
25	0.1547	529	16	0.1288	434	16	0.1288	434	12	0.074	37	12	0.074	37
28	0.4857	529	18	0.1901	434	18	0.1901	434	11	0.111	37	11	0.111	37
31	0.4968	529	22	0.233	471	22	0.233	471	15	0.151	37	15	0.151	37
			26	0.2697	471	26	0.2697	471	18	0.191	37	18	0.191	37
			29	0.2883	241	27	0.3734	230	22	0.2339	471	22	0.2339	471
			30	0.3928	241	30	0.3919	230	26	0.2706	471	26	0.2706	471
			31	0.4952	471	31	0.4943	471	27	0.3743	230	29	0.2892	241
									30	0.3928	230	30	0.3937	241
									31	0.4952	471	31	0.4961	471

Table 4.4.9 Network 4: Car Path for Car Only Trips

PATH 1		
NODES	CULM. TIME (Minutes)	FLOW
27		
31	0.0273	1,000
30	0.0455	1,000
35	0.0691	1,000
34	0.0873	1,000
33	0.1256	1,000

Table 4.4.10 Network 4: Car Path for Combined Trips

PATH 1		
NODES	CULM. TIME (Minutes)	FLOW
27		
31	0.0273	1,000
30	0.0455	1,000
35	0.0692	1,000
34	0.0874	1,000
33	0.1257	1,000

Table 4.4.11 Network 4: Truck Path for Truck Trips

PATH 1		
NODES	CULM. TIME *(Minutes)*	FLOW
37		
32	0.0191	250
31	0.0458	250
30	0.064	250
35	0.0876	250
34	0.1058	250
39	0.144	250
48	0.1775	250

Table 4.4.12 Network 4: Truck Path for Combined Trip

PATH 1		
NODES	CULM. TIME *(Minutes)*	FLOW
37		
32	0.0191	250
31	0.0458	250
30	0.064	250
35	0.0877	250
34	0.1059	250
39	0.1441	250
48	0.1776	250

Table 4.4.13 Network 5: Car Path for Car Only Trips

PATH 1			PATH 2			PATH 3		
NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW
13			13			13		
19	0.0505	476	19	0.0505	476	14	0.0574	524
20	0.0931	399	25	0.0839	77	15	0.1148	524
21	0.1357	399	26	0.1173	77	55	0.1351	524
54	0.1539	399	27	0.1507	77	54	0.1533	524
53	0.1721	923	53	0.1682	77	53	0.1715	923
52	0.1903	1,000	52	0.1864	1,000	52	0.1897	1,000
51	0.2085	1,000	51	0.2046	1,000	51	0.2079	1,000
50	0.2267	1,000	50	0.2228	1,000	50	0.2261	1,000
49	0.2449	1,000	49	0.241	1,000	49	0.2443	1,000

Table 4.4.14 Network 5: Truck Path for Truck Only Trips

PATH 1			PATH 2			PATH 3		
NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW
13			13			13		
19	0.0608	123	19	0.0608	123	14	0.0645	127
20	0.1055	101	25	0.0941	22	15	0.129	127
21	0.1502	101	26	0.1274	22	55	0.1486	127
54	0.1683	101	27	0.1607	22	54	0.1668	127
53	0.1865	228	53	0.1862	22	53	0.185	228
52	0.2047	250	52	0.2044	250	52	0.2032	250
51	0.2229	250	51	0.2226	250	51	0.2214	250
50	0.2411	250	50	0.2408	250	50	0.2396	250
49	0.2593	250	49	0.259	250	49	0.2578	250

Table 4.4.15 Network 5: Passenger Car Path for Combined Trips

PATH 1			PATH 2			PATH 3			PATH 4			PATH 5			PATH 6		
NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW
13			13			13			13			13			13		
19	0.1054	473	19	0.1054	473	19	0.1054	473	14	0.1316	527	14	0.1316	527	14	0.1316	527
20	0.1459	243	20	0.1459	243	25	0.1443	230	15	0.1948	372	20	0.1532	155	20	0.1532	155
26	0.1793	54	21	0.1993	344	26	0.1828	230	55	0.2161	372	26	0.1866	54	21	0.2066	344
27	0.2255	284	54	0.219	344	27	0.229	284	54	0.2343	372	27	0.2328	284	54	0.2263	344
53	0.2442	284	53	0.2372	716	53	0.2477	284	53	0.2525	716	53	0.2515	284	53	0.2445	716
52	0.2624	1,000	52	0.2554	1,000	52	0.2659	1,000	52	0.2707	1,000	52	0.2697	1,000	52	0.2627	1,000
51	0.2806	1,000	51	0.2736	1,000	51	0.2841	1,000	51	0.2889	1,000	51	0.2879	1,000	51	0.2809	1,000
50	0.2988	1,000	50	0.2918	1,000	50	0.3023	1,000	50	0.3071	1,000	50	0.3061	1,000	50	0.2991	1,000
49	0.317	1,000	49	0.31	1,000	49	0.3205	1,000	49	0.3253	1,000	49	0.3243	1,000	49	0.3173	1,000

Table 4.4.16 Network 5: Truck Path for Combined Trips

PATH 1			PATH 2			PATH 3			PATH 4			PATH 5			PATH 6			PATH 7		
NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW	NODES	CULM. TIME (Minutes)	FLOW
13			13			13			13			13			13			13		
19	0.1857	122	19	0.1857	122	19	0.1857	122	19	0.1857	122	14	0.191	128	14	0.191	128	14	0.191	128
20	0.2274	64	20	0.2274	64	25	0.2247	58	25	0.2247	58	15	0.2718	93	20	0.2251	35	20	0.2251	35
21	0.2849	77	26	0.2608	23	31	0.258	4	26	0.2627	54	55	0.2927	93	21	0.2826	77	26	0.2585	23
54	0.3039	77	27	0.3123	77	32	0.2913	4	27	0.3142	77	54	0.3109	93	54	0.3016	77	27	0.31	77
53	0.3221	170	53	0.3309	77	33	0.3246	4	53	0.3328	77	53	0.3291	170	53	0.3198	170	53	0.3286	77
52	0.3403	246	52	0.3491	246	52	0.3421	4	52	0.351	246	52	0.3473	246	52	0.338	246	52	0.3468	246
51	0.3585	250	51	0.3673	250	51	0.3603	250	51	0.3692	250	51	0.3655	250	51	0.3562	250	51	0.365	250
50	0.3767	250	50	0.3855	250	50	0.3785	250	50	0.3874	250	50	0.3837	250	50	0.3744	250	50	0.3832	250
49	0.3949	250	49	0.4037	250	49	0.3967	250	49	0.4056	250	49	0.4019	250	49	0.3926	250	49	0.4014	250

Table 4.4.1 shows a pair of passenger car paths when applying 1000 car trips from Centroid 1 to Centroid 4. The two paths have identical travel time of 0.0567 with 555 car flow allocated on Path 1. The 250 truck trips from Centroid 1 to Centroid 4 utilize the same two paths as the passenger cars as shown in Table 4.4.2. When the passenger car trips and truck trips are combined in the same network, both paths have almost the same travel time (due to the computer program's convergence measure) for the car trips, and for the truck trips (See Tables 4.4.3 - 4.4.4). However, the paths flow allocations are slightly changed, which demonstrates the interaction between the cars and the trucks when they simultaneously use the network.

Table 4.4.5-4.4.8 shows the results from the application of the same four scenarios on network 3. Network 3 is a much more complicated network compared with Network 1. The car trips are from Centroid 1 to Centroid 3, and all the cars use the path along the outer edge of the network. The truck trips are from Centroid 2 to Centroid 3. Three paths with almost identical travel time are used by truck trips, where one of them use a section of the car trip path and attract 595 trips out of total 100 trips, and the other two paths have almost same links. When the passenger car and truck trips are merged, the trucks cause congestion on part of the passenger car path, and therefore 263 out of 1000 car trips divert to the other two paths with similar travel time as the original one. The congestion by additional car trips cause longer travel time for the trucks on their original three paths, so that the two new paths become attractive and the diverted 37 trucks on the new paths balance the travel time over 5 paths, which is around the neighborhood of 0.495 minutes (See Table 4.4.8).

In network 4 , the car trips origin is Centroid 2 and the destination is Centroid 8 respectively. Similarly, the origin for the trucks is Centroid 3 and the destination is Centroid 7. Both passenger cars and trucks have one dominate path in their own trips assignment (Table 4.4.9 and Table 4.4.10). Since the car path and truck path use different links , when the trips are combined, a very small impact is observed, a slight increase in the travel time of the shared links takes place on both car trips and trucks. No new paths are generated in this case.

Network 5 represents a highway network in an suburban area, where a major freeway is accompanied with a pair of parallel service roads. Three paths with identical travel times are used by both passenger cars and trucks under the individual traffic assignments are conducted separately (Table 4.4.13 and Table 4.4.14). When their trips are combined together, under the same traffic assignment, three more identical new paths for the passenger car trips, and one more new path is created by truck trips. (See Table 4.4.15 and Table 4.4.16).

In general, the interaction between passenger cars and trucks, caused by sharing the right of the way simultaneously in the network, primarily increases the congestion, diverts both car and truck traffic to other new paths, though the magnitude of the impact to passenger cars and trucks is different.

From the results of these experiments, it can be confirmed that the diagonalization algorithm does converge to an equilibrium flow pattern. It is also confirmed the travel times in all paths are identical for each user class separately, which confirms that this is a UE traffic pattern (Tables 4.4.1 - Table 4.4.16.). These results are very important for the TCTNDP, since they provide confidence that the solution of the traffic assignment at each iteration is a UE traffic pattern.

CHAPTER 5

DEVELOPMENT OF THE COMBINED SIMULATED ANNEALING AND TABU SEARCH STRATEGY (SA-TABU)

This chapter presents combined simulated annealing and tabu search strategy (SA-TABU) to solve the two classes of users equilibrium transportation network design problem (TCNDP). The search strategy developed identifies the best combination of links to be expanded and their three traffic operation options for capacity improvement within a feasible budget constraint such that the system wide total UE travel time is minimized. The problem was formulated as an integer, nonlinear programming problem, where the nonlinear two classes of users link travel time function (Formulas 4.2.4, 4.2.5) was used.

5.1 General Background of the Combined Simulated Annealing and Tabu Search Algorithm in this Study

The most important feature of the simulated annealing algorithm is that the acceptance of the new solution state is probabilistic. The new solution state that improves the objective function, is unconditionally accepted, and on the other hand, the probability of accepting the non-improved solution state increases with the quality of the new solution states-the better objective function values, the higher quality of the new solution state is, and decreases with the progress of the search. In general, the simulated annealing algorithm tends to accept all the moves - from the current solution state to a new solution state, in the early stage of the search, and as the search goes on, it starts to focus on some good solution space and finally becomes a local optimum search strategy approach.

As mentioned in Section 2.6.1, the original version of the simulated annealing algorithm requires that the generation of the new solution state should be randomized and the decreasing rate of the move acceptance probability should be very small in order to reach "annealing" status. Such procedure is not suitable for large scale network design problems, since the computing time of the solution state objective function value is extremely high, and the number of possible solution states is prohibitively high.

By introducing heuristic information, the generation of the new solution state would be more informed, and the simulated annealing process would only search in a high quality solution state region. Therefore, it has the potential to dramatically improve the efficiency of the algorithm, though the risk of not finding the global optimum may also be increased. However, with the proper design of the heuristic evaluation function (HEF), an efficient simulated annealing algorithm could be designed to find satisfactory near optimal solutions.

The historical information may be considered as one of the HEF variables. The information about the impact of each link or each of the three options from the past experience, which is gained throughout the search process, provides valuable information. Given the significant computational expense of the two classes of users user equilibrium traffic assignment (TCUEA) procedure, it becomes a necessity to seek sufficiently good solutions within a reasonable number of iterations of the

TCNDP. Including the historical information as part of the HEF variables may greatly increase the algorithm's efficiency. A "noise" or a random error can be added to the HEF to provide variability of the candidate solution states, as well as to fulfill the randomization property of the original simulated annealing algorithm.

Tabu search is considered a rather more aggressive heuristic search methodology. The moves produced by the tabu search algorithm relies more on the HEF, regardless of the quality of the new solution state. One of its most important features is the use of the tabu list, which reduces the risk of cycling. Early numerical experiments show that cycling and reoccurrence of the solution states greatly affect the effectiveness of the simulated annealing with the use of an HEF. The more informed the HEF is, the higher the risk of cycling. Therefore, the combination of the convergent characteristics of simulated annealing and the reduction of the risk of cycling through tabu search provided the rationale for developing a combined simulated annealing / tabu search strategy (SA-TABU) to solve the TCNDP.

The following sections present the basic elements of SA-TABU approach developed for this study.

5.2 Heuristics Based Combined Simulated Annealing and Tabu Search (SA-TABU) Strategy

The basic elements of the SA-TABU search strategy are: type of moves, HEF and the search strategy itself. These elements are presented in the following subsections:

5.2.1 Type of Moves

A move is defined as the basic mechanism which achieves a change from the current solution state to a new solution state. A solution state in the TCNDP refers to the network's configuration.

The network design problem decision variables are 0-1 integer numbers. A pair of two sets can be used to present a solution state. One set includes all the variables with values equal to one (expanded links), which is defined as the Solution-1 set. The other set contains the variables with zero value (non-expanded links), and it is defined as the Solution-0 set.

The new solution state can be generated by exchanging the elements in two sets(e.g. some variables change from 0 to 1 and some from 1 to 0). At the same time the budget constraint (Formula 3.1.2) must be satisfied to ensure the feasibility of the new solution state. The add/drop move, knapsack move, and random perturbation are the most frequently methods used to produce a new solution state. The knapsack type move generates the new solution state by solving a linear integer programming problem. The random perturbation method randomly swaps a number of elements between the two sets to generate a new solution state, and is widely used in bipartite graph problems and travel salesman problems. The add/drop type of move, presented in Figure 5.2.1, exchanges source element(s) belonging to Solution 1 set which have the least HEF value(s) with source element(s) belonging to Solution 0 set which have the highest HEF value(s). Similarly, as before, the new solution state must satisfy the budget constraint.

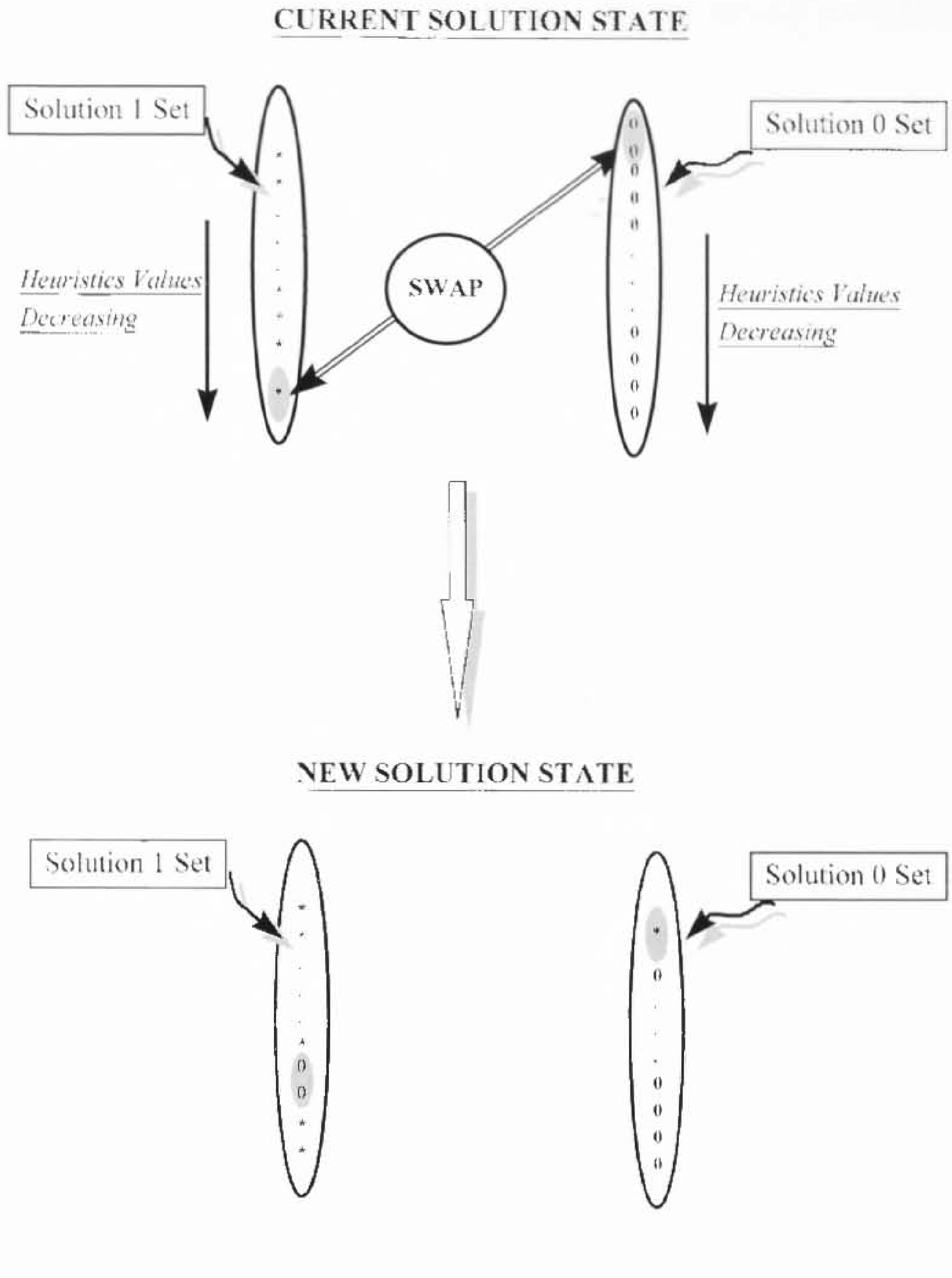


Figure 5.2.1 Illustration of the Add/Drop Type Move

The knapsack move and add/drop move were applied by Mouskos (1990) to solve the single class network design problem (SCNDP) utilizing tabu search. Balakrishnan, et. al.(1989) employed the add/drop type move for the solution to the large scale uncapacitated network design application with dual ascent procedure, and Janson et. al.(1983) also used the add/drop type move in a network design application for highway improvements on a real highway network.

Although the knapsack move can immediately identify a new solution state taking into consideration the HEF values for all the variables of the problem set, it has a higher computational cost by requiring the solution of a linear integer programming problem. Its effectiveness is not yet clear. However, it can be stated that it can provide a good starting solution, and it can redirect the search for a local optima, thereby diversifying the search (Mouskos, 1991). The add/drop type move generates the new solution state from the current solution state by a small perturbation and heavily relies on the HEF 's information. It also performs like a neighborhood search. Furthermore, it has low computational cost and feasible new solution states could be found easily. Another attractive characteristic of the add/drop type move is that it can provide valuable historical information of the contribution of each move to the current solution state from the previous one. Based on these characteristics, the add/drop type move is used in this study.

5.2.2 Heuristic Evaluation Function (HEF)

The available budget in the network design problem is usually small and the degree for changing the network configuration is rather limited, so the redistribution of the network flow usually does not significantly change most of the links HEF are primarily based on the link flow. If some critical link flow patterns happen to result in low HEF values, they may not be considered for capacity expansion during the whole search procedure process. In addition, other information which emerge during the search such as the solution state objective function values may not be fully utilized.

At present, most of the heuristics functions used are non solution specific. For example, in Mouskos (1990), the new solution state projecting the new configuration of the network results in the redistribution of the network flow and the change of each link heuristics values such as V/C ratio, speed and travel time values, and then these new heuristics functions values are used to generate the new solution state. Thus, in this approach, the generation of the new solution state only depends upon the current solution state link flow pattern . A solution-specific HEF is developed in this study.

A simplified version of the composite HEF used in this study is as follows:

$$H_i = V_i / C_i + F_1 \times rand(0,1) + BH_i / (MAX(BH_i, \forall i)) \tag{5.2.1}$$

H_i : HEF value for Link i ,

V_i / C_i : Link i 's V/C ratio,

F_1 : Random variable expanding factor,

BH_i : Link i 's historical contributions to the objective function, this component is termed as LCOF.

The main characteristics of the composite HEF developed in this study are: The inclusion of the V/C ratio which captures the link's current traffic flow status; The second component introduces a random variable to the HEF which acts as an error term. In essence, the HEF with the inclusion of this random "error" term becomes a stochastic HEF from a deterministic one. The use of the current value of the solution state and the historical solution state information in the third term (LCOF) provides an additional element in the search that rewards variables which performed well in the past and penalizes those that did not.

The maintenance of the LCOF is shown in Figure 5.2.2. The LCOF values are only updated for the links whose decision variable values are changed (either enter or exit the Solution-1 set). For example, if the new solution state has better objective function value than the current solution state, the links which were changed from solution 1 set to the solution 0 set (we refer to the links that have such change as dropped links) are believed to be inferior to the links which were changed to the solution 1 set from solution 0 set (we refer to the links that have this change as added links) and thus the dropped links are penalized in their heuristics values while the added links heuristics function values are credited due to their better contribution to the objective function. On the other hand, if the new solution state is worse than the current solution state, the added links will be penalized and the dropped links will be credited (See Section 5.3).

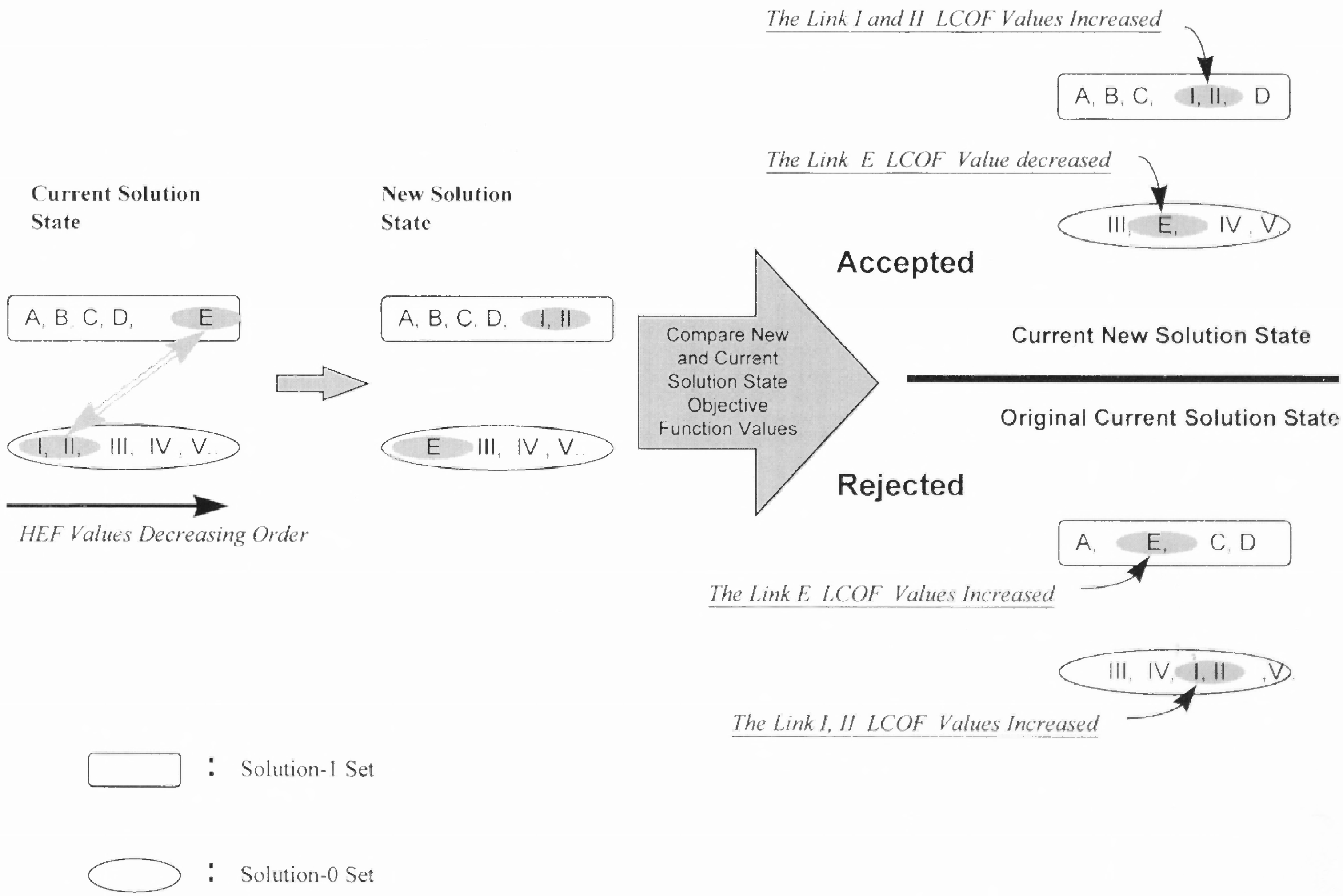


Figure 5.2.2 Updating the LCOF Values

The inclusion of the above mentioned elements into the HEF prompted the use of the term solution-specific HEF. The solution-specific HEF originates from the concept of the neural networks. In the neural network process, the "neuron" is "trained" or adapted by the feedback of the objective function value derived from the previous "neuron". As the search process continues, the links which contribute most to the network's objective function value form a cluster of links in the Solution-1 set of the final solution.

More details about the solution specific HEF used in this study can be found in Section 5.4.

5.2.3 Search Strategy

The proposed algorithm for this study borrows the main structure of the simulated annealing algorithm, using tabu type of moves and tabu lists to reduce cycling and to local optimum, in an attempt to develop an efficient, robust search strategy for the solution of the TCNDP.

Preliminary experiments of the above search strategy indicated that cycling was encountered very frequently. This resulted from the fact that after a number of iterations, some links HEF values have significant lead over others due to their positive impact on the improvement of the objective function value, which forces them back in the solution if they are dropped. To avoid cycling, tabu lists are utilized to inactivate certain moves for a few iterations and let the search spread over a wider solution space.

In the algorithm, developed the tabu list contains the reverse of the moves most recently made. For example, if the Link A is just dropped from the Solution-1 set to the Solution-0 set (e.g. $y_a=1 \Rightarrow y_a=0$), the reverse move is the one that the Link A is added to Solution-1 set from Solution-0 set (e.g. $y_a=0 \Rightarrow y_a=1$). The implementation of the tabu list is implemented through the use of the link's heuristic values. After Link A is dropped from Solution 1 set and resides in Solution-0 set ($y_a=0$), Link A is assigned with the lowest HEF value in the Solution-0 set, and thus Link A has the smallest probability to be added up to Solution 1 set in next few moves. When the tabu period (number of iterations) for the Link A is expired, it is reassigned back to its original HEF value prior to entering the tabu list.

The next section presents the basic steps of the SA-TABU search strategy developed in this study.

5.3 General Procedure of the Heuristic Based Simulated Annealing and Tabu Search Strategy (SA-TABU).

In the previous section, the basic components of the SA-TABU search strategy were introduced. Following, the basic steps of the SA-TABU search strategy to solve the TCNDP are presented (See also Figure 5.4.1).

STEP 1 The program starts from an initial solution state $\underline{y}_0$ with the following initial values of the parameters : i) initial scheduling control parameter $\underline{C}_0$ (usually referred as "temperature"); ii) initial maximum Markov chain length L_0, and maximum number of iterations $Iter_{max}$. Compute y_0 objective function value f_0 (total UE network travel time)

STEP 2: Determine the starting control parameter C_k .

STEP 2.0 Let C_k=C_0.

STEP 2.1 Use the random generator to generate the trial solution state y_t from the current solution state y_c and compute its objective function value f_t (current total UE network travel time)

STEP 2.2 If $\exp((f_t - f_0)/C_k) >$ Random (0,1), then accumulate the number of transition trials by 1, otherwise accumulate the number of the non-transition trials.

STEP 2.3 After 20 iterations of procedure STEP 2.1 to STEP 2.2, if the number of transition trials is greater than 16 (Acceptance rate is greater than 0.8), then go to STEP 3, otherwise let $C_k = 2 \times C_k$, and reset the number of transition trials and non-transition trials to 0, before going back to STEP 2.1.

STEP 3 Use the trial solution state generator to generate the new trial solution state y_t from the current solution state y_0 (This is further explained later in the section.)

STEP 4 Compute the network total travel time f_t for the new trial solution state y_t.

STEP 5 If $\exp((f_t - f_3)/C_k) >$ Random (0,1), then go to STEP 6 , otherwise increase the number of non-transition indicator Nochg by 1 and go to STEP 9.

STEP 6 Make the move and set Nochg=0. The current solution state is overwritten by the new trial solution state. $y_0 \leftarrow y_t, f_0 \leftarrow f_t$.

STEP 7 Accumulate the Markov chain length L by 1, and if $L < L_0$, then go to STEP 9

STEP 8 Decrease the control parameter C_k and increase the Maximum Markov chain length L_0 and let L=0.

STEP 9 Update the link's HEF values.

STEP 10 Increase the number of iteration indicator Iter by 1. If Iter is equal to $Iter_{max}$ or Nochg reaches the maximum number of no change parameter , Stop, otherwise Go to Step 3.

The generation of the trial (STEP 3) solution state is illustrated in Figure 5.3.2. The following steps present how to use the add/drop move to generate a feasible trial solution state utilizing the link heuristics HEF values:

STEP 3.1 Generate Solution 1 set and Solution 0 for the current solution state and update the link HEF values.

STEP 3.2 Sort the Solution 1 set and Solution 0 set links based on the current HEF values in decreasing order.

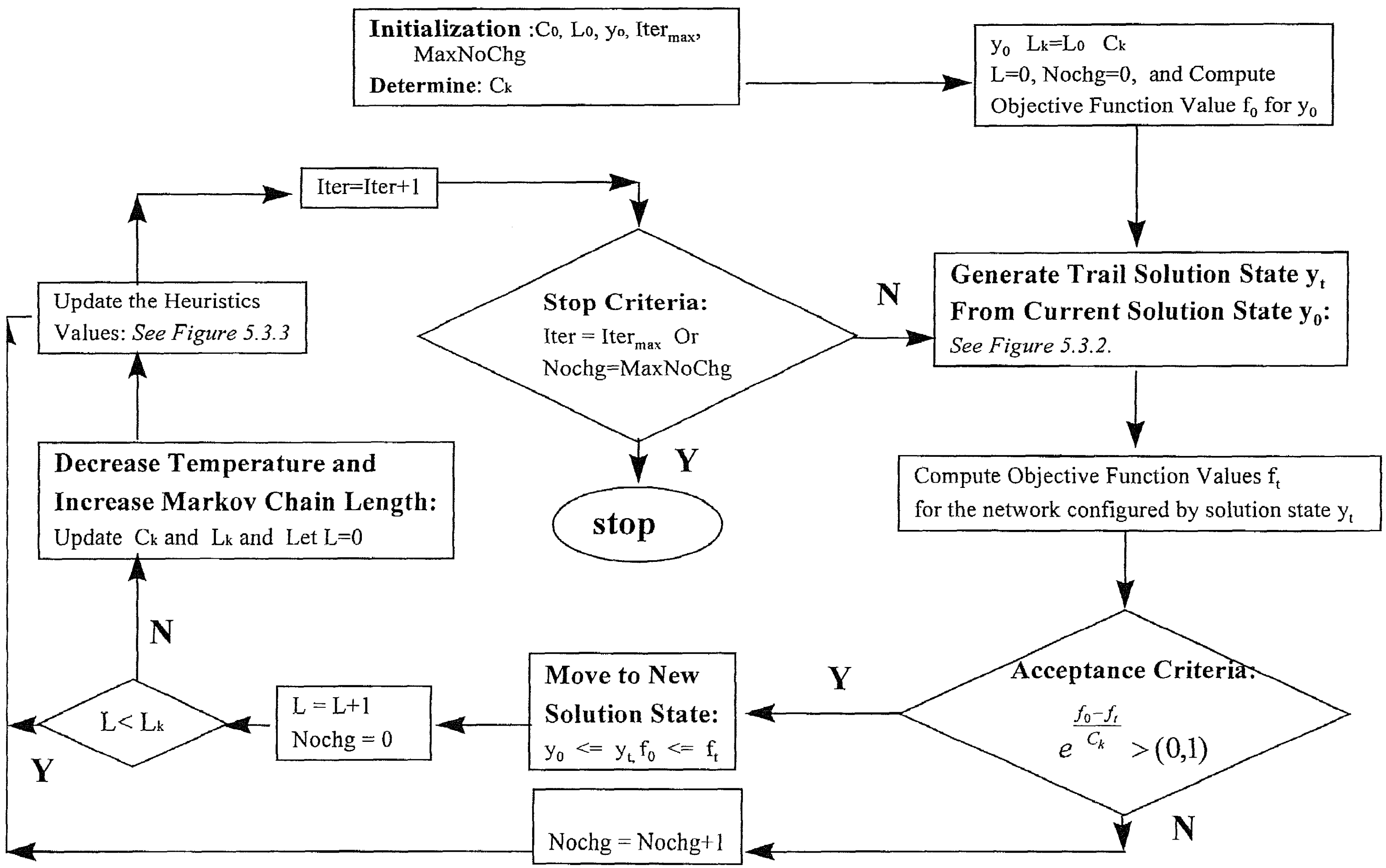


Figure 5.3.1 Flowchart of SA-TABU Search Strategy

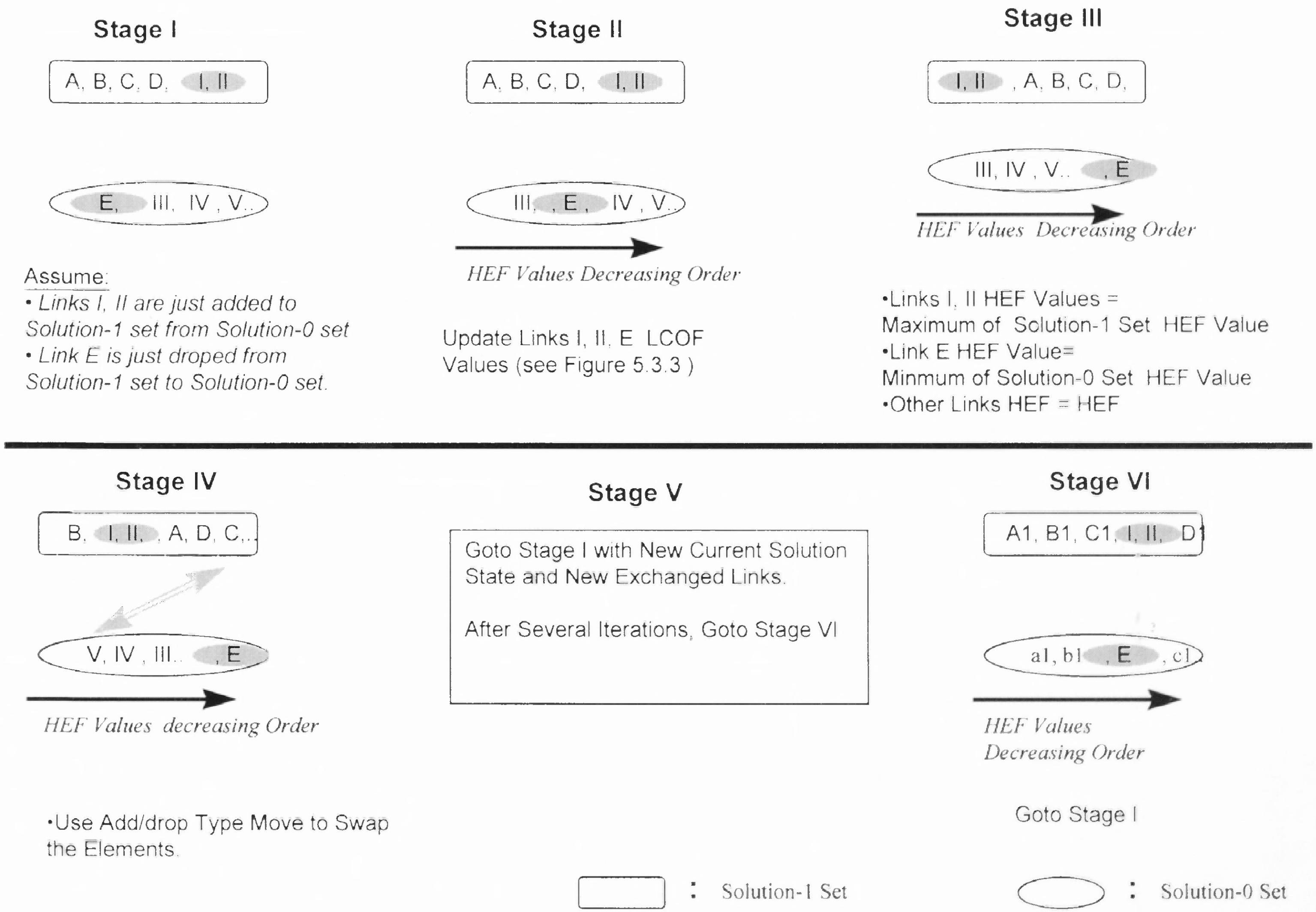


Figure 5.3.2 Illustration of Generating a New Trial Solution State

STEP 3.3 Extract the link with the least HEF value in the Solution 1 set and retrieve its budget value b1. Extract the link with the highest HEF value in the Solution 0 set and retrieve its budget value b0.

STEP 3.4 If $b1 - b0 > 0$, let $b1 = b1 - b0$, and set the value of the link extracted from Solution 1 set to 0 and from Solution 0 set link to 1, otherwise go to STEP 3.7.

STEP 3.5 Extract the link with next highest HEF value in the Solution 0 set and retrieve its budget value b0.

STEP 3.6 If $b1 - b0 > 0$, let $b1 = b1 - b0$, set the link value to 1 and go to STEP 3.5, otherwise Stop.

STEP 3.7 Extract the link with next highest HEF value in the Solution 0 set and retrieve its budget value b0.

STEP 3.8 If $b1 - b0 > 0$, let $b1 = b1 - b0$, and set the link value to 1. Go to STEP 3.7, otherwise go to STEP 3.9.

STEP 3.9 Extract the link having the next least HEF value from Solution 1 set, and retrieve the budget value b2, and let $b1 = b1 + b2$.

STEP 3.10 Extract the link with the highest HEF values in the Solution 0 set and retrieve its budget value b0.

STEP 3.11 If $b1 - b0 > 0$, let $b1 = b1 - b0$, and set the value of the link extracted from Solution 1 set to 0 and Solution 0 set link to 0, otherwise go to Step 3.10.

STEP 3.12 Extract the link with the next highest HEF value in the Solution 0 set and retrieve its budget value b0.

STEP 3.13 If $b1-b0>0$, let $b1=b1-b0$, let the link value to 1 and go to Step 3.12, otherwise Stop.

The procedure and formula for updating the link LCOF values is outlined in Figure 5.3.3. Assuming that :

i) C_0 is the initial processing control parameter (called "temperature") and C_c is the current control parameter;

ii) f_c, f_b, f_t is the objective function value of the current solution state, solution state with best objective function value, and the present trial solution state respectively;

iii) The variables set E, A is a set of variables in the Solution 1 set of the current solution state and in Solution-0 set in the trial solution state, and B is vice versa;

iv) $heuristic_i$, v_i/c_i is the ith link LCOF value and V/C ratio, and $fheur_i$ is the ith link HEF value to be used for generating a trial solution set;

v) F1 is the random number expanding factor.

Thus links HEF updating procedures is as:

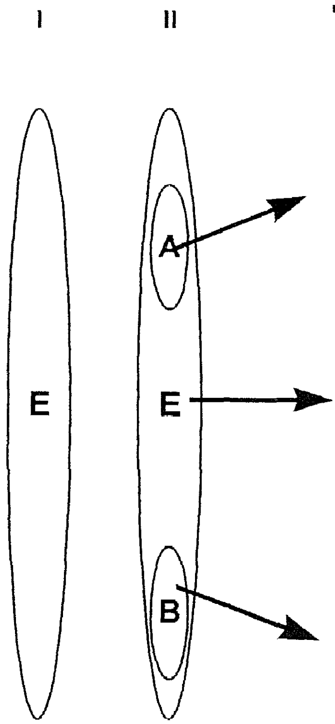
$$fheur_i = V_i / C_i + F_1 \times rand(0,1) + heuristic_i / (MAX(heuristic_i, \forall i)) \qquad (5.3.1)$$

if $f_t < f_b$, we have:

$$cbest = \frac{C_0 - C_c}{C_0} + 1 \qquad (5.3.2)$$

other wise

$$cbest = 1 \qquad (5.3.3)$$



	$f_t < f_c$	$\exp((f_c - f_t)/T_c) > Random(0,1)$	Otherwise
A	Drop Penalize Factor: $0 < dpp < 1$ $Cbest = (C_0 - C_c)/C_0 + 1$ $LCOF =$ $(LCOF) * Cbest * dpp$	Same as Left Except: $Cbest = 1.0$	Drop Credit Factor: $dpc > 1$ $Cbest = 1.0$ $LCOF =$ $(LCOF) * Cbest * dpc$
E	**No Change**	**No Change**	**No Change**
B	Add Credit Factor: $adc > 1$ $Cbest = (C_0 - C_c)/C_0 + 1$ $LCOF =$ $(LCOF) * Cbest * adc$	Same as Left Except: $Cbest = 1.0$	Add Penalize Factor: $0 < adp < 1$ $Cbest = 1.0$ $LCOF =$ $(LCOF) * Cbest * adp$

Note:

I: Current Solution State II:New Solution State

E: Variable Set A: Set of Variables Dropped From Solution-1 Set to Solution-0 Set and B is vice versa.

f_t: New Solution State Objective Function Value f_0: Current Solution State Objective Function Value

C_0: Initial Temperature C_c:Current Temperature

Figure 5.3.3 Diagram of Link Updating LCOF Values in SA-TABU Approach

If $\exp((f_t - f_0)/C_k) >$ Random $(0,1)$, then:

$$\text{(i) } heuristic_a = dpp \times cbest \times heuristic_a, \qquad \forall a \in A, \quad 0 < dpp < 1 \qquad (5.3.4)$$

$$\text{(ii) } heuristic_b = adc \times cbest \times heuristic_b, \qquad \forall b \in B, \quad adc > 1 \qquad (5.3.5)$$

otherwise, we have:

$$\text{(i) } heuristic_a = dpc \times cbest \times heuristic_a, \qquad \forall a \in A, \quad dpc > 1 \qquad (5.3.6)$$

$$\text{(ii) } heuristic_b = adp \times cbest \times heuristic_b, \qquad \forall b \in B, \quad 0 < adp < 1 \qquad (5.3.7)$$

As described in the previous section and also in Figure 5.3.2, in order to prevent cycling, the link HEF values are updated based on the tabu condition to prevent the reversal of the most recently made moves. The elements which are dropped from Solution-1 set to Solution-0 set are temporarily assigned with the minimum HEF value in the Solution-0 set. These elements have very small probabilities to be selected back to the Solution-1 set again when a new trial solution state is generated. However, after several iterations (the length of the tabu list) they are reassigned to their origin HEF values and they become candidates for inclusion to the new solution. Similarly, the elements that change from Solution-0 set to the Solution-1 set are assigned very high temporary HEF values (for the length of the tabu list). The use of these two tabu lists in the Solution-1 set and Solution-0 set respectively, force the search strategy to select new elements to either enter or drop to/from the Solution-1 set.

5.4 Implementation of the SA-TABU Search Strategy

In this study, several different versions of the SA-TABU search strategy methods were developed. The next section describes the computer program implementation of the SA-TABU search strategy developed. The program is written in Fortran 77 language computer code and is compiled by FORTRAN SPAR Compiler in Sun workstation UNIX operating system environment.

5.4.1 Main Program (Figure 5.4.1-5.4.2)

This program consists of two main steps. The first step determines the initial control parameter value t0, or "temperature". The second step is the main process of the SA-TABU search strategy. These two steps are very similar, and they both use the basic procedures of the simulated annealing approach, where the primary difference is in the generation of the trial solution state. The random swapping trial solution state generation strategy is used for the first step, while the trial solution state generation method utilizing the comprehensive heuristic based add/drop type move, with the activation of the tabu list takes place in the second step.

The main program starts with initialization of several parameters such as the "penalty" and "credit" parameters for the link HEF values, Markov chain length, control parameter decreasing rate and maximum number of iterations. These parameters differentiate the various versions of the search strategies developed. The network data such as the origin destination (O-D) trip table, network configuration and link attribute data are read within the ***INITUE*** subroutine. The ***UE*** subroutine solves the user equilibrium (UE) traffic assignment to obtain the initial traffic flow for the initial network. Then the ***Init_grade*** procedure is called to get the initial link HEF values (score), which is the function of the initial link congestion factor. The initial solution state (y0) is generated by subroutine ***Initial_y***. The subroutine ***Compute_Y***, which is a modified version of the combination of the ***UE*** subroutine and other subroutines is called to compute the initial solution state total travel time(f0).

The "temperature" - control parameter is initialized (See Figure 5.4.1) in a way such that the starting "temperature" would be close to the desired initial "temperature" in most of the scenarios. The subroutine ***perbation1*** is activated to generate the trial solution state (newy), and the ***Compute_y*** subroutine computes the objective function values (f1), and the simulated annealing acceptance criteria are used to determine whether or not to accept the trial solution state. The preceding procedure is repeated for 20 times. If the trial solution state acceptance ratio is less than 80%, the "temperature" value is doubled. This procedure continues until the acceptance ratio exceeds 80 %, and then the current "temperature" is accepted as the initial "temperature" for use in the second step of the search.

Having a value for the initial "temperature", the SA-TABU procedure is activated. The current solution state and value (yn, f0) are stored as the best solution state and value (fbest, ybest), respectively. The tabu list is initialized as an empty set. The subroutine ***tgrade_perbation1*** is called to generate the trial solution state (newy) by the combined method and the ***compute_y*** subroutine computes its objective

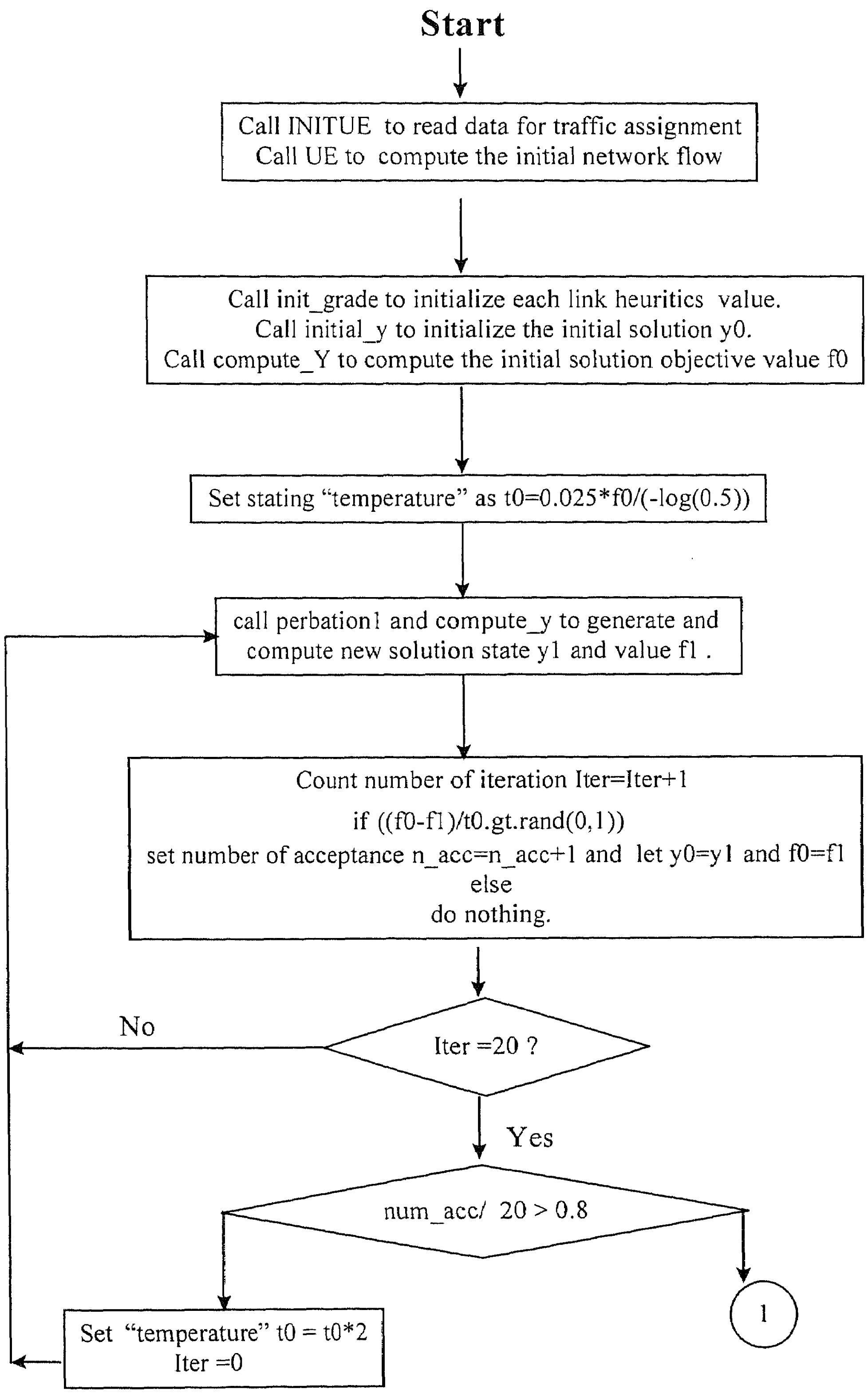


Figure 5.4.1 Main SA-TABU Program Flow Chart (1/2)

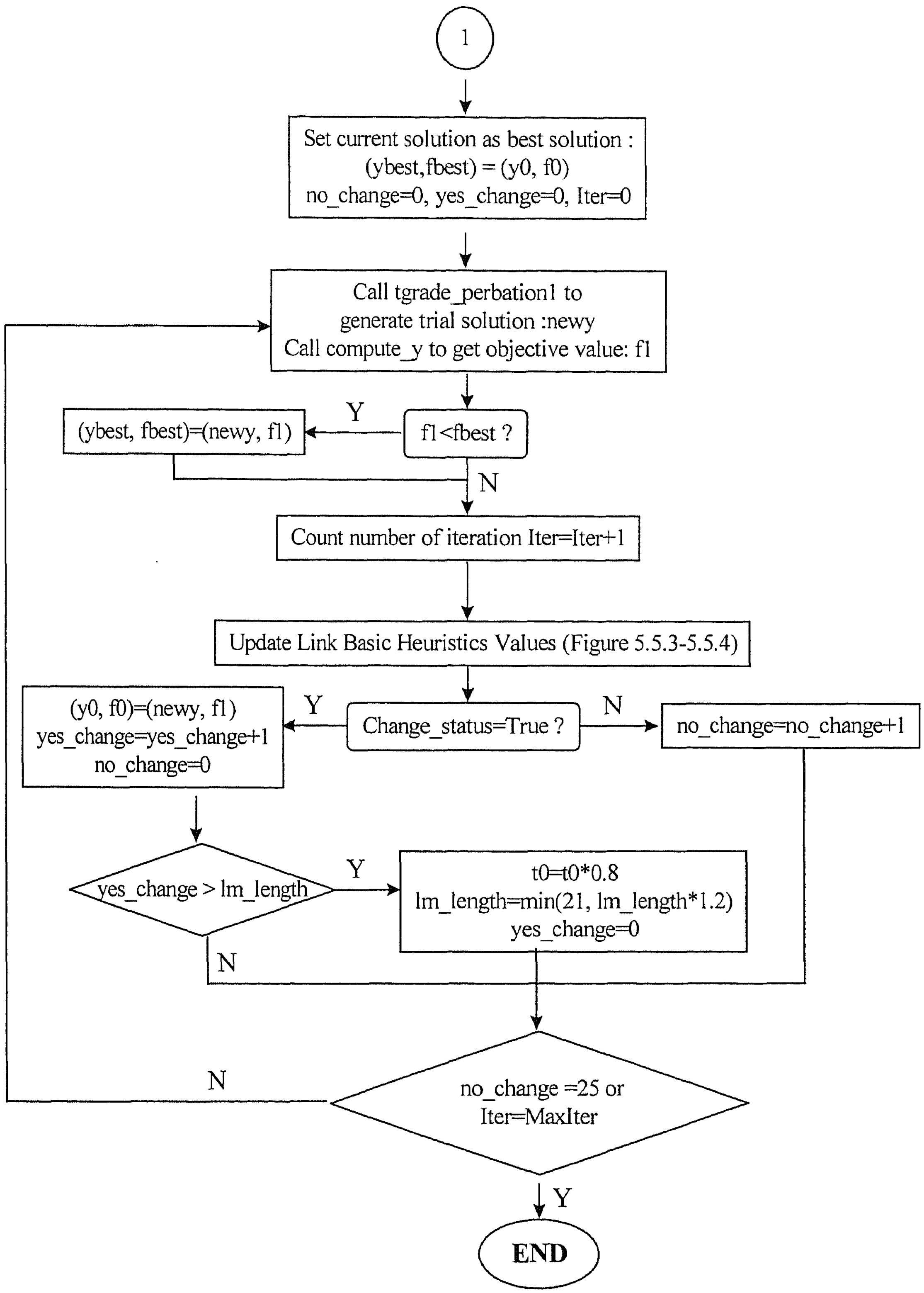


Figure 5.4.2 Main SA-TABU Program Flow Chart (2/2)

function value (f1). The best solution (fbest, ybest) is updated if the solution (newy,f1) is the best solution state. The acceptance criteria is used to judge if the (newy, f1) is accepted as the next current solution state. Then, the updating of the HEF values (Figure 5.4.3 - 5.4.4) is conducted.

The value impact factor (tini) is used to consider the percentage improvement of the trial solution state against the current solution state. The higher the improvement,((fo-f1)/f0) is, the greater the impact factor is. The "temperature" impact factor (a) is used to count the difficulty of accepting the trial solution state. As the "temperature" decreases, the quality of the accepted trial solution state increases, and the "temperature" impact factor awards the higher quality solution state a better "credit" by multiplying the related link HEF values by a. Thus, factor "a" is increased based on the decrease of the current "temperature" "t0" by a factor of (1+(tini-t0)/t0).

The array score stores the link HEF values for non tabu links and the temporary HEF values for the tabu links. The array hiscore has two functions. The first function is to record the link's tabu status, and the other is to store the origin HEF values for the links in the tabu list. If a link's hiscore value is positive, it implies that the link is currently in the tabu list. The hiscore value for a link in tabu list is the link's origin HEF value and the variable score for this link is assigned a number, a temporary HEF value, which causes this link to have a very low probability to be considered for swapping. The array last_add records the most recent added link into the solution and the corresponding added link's value is changed from 0 to 1, which implies that the link is added to the Solution-1 set from the Solution-0 set. The array last_drop records the link that being dropped most recently from the Solution-1 set to the Solution-0 set. If, for a link residing in the tabu list, the difference of the current iteration and its last_add or the difference of the current iteration and last_drop reaches the tabu list length, the links residing in the tabu list exit by setting score(i) to its basic HEF value hiscore(i) and hiscore(i) is set to -1.

The following scenarios are applied to links equal to 1 in the current solution state and equal to 0 in the trial solution state, or dropped out from the Solution 1 set to the Solution 0 set, and the trial solution state has been accepted (change_status=true): i) if the trial solution state is better than the current solution state, the link's LCOF value is penalized more by the "temperature" impact factor; ii) if the trial solution state is the overall best solution state so far, the link's LCOF value is penalized more by the best solution factor. However, in an extreme case, the tabu links may still participate in the swapping, since the final HEF value (Formula 5.3.1) is stochastic.

Therefore, if the link is in the tabu list and drops from Solution-1 set to Solution-0 set, the link would immediately exit the tabu list and the LCOF value would be factored by the above scenarios. The non-tabu link would enter the tabu list by setting hiscore(i) to the origin HEF value score(i) and the score(i) to the minimum value of all of the Solution-0 set's links, which is a temporary HEF value, and then set the last_drop to the current iteration number.

However, if the trial solution state is not accepted (change_status=false), the link's basic HEF values are increased by the drop_credit factor.

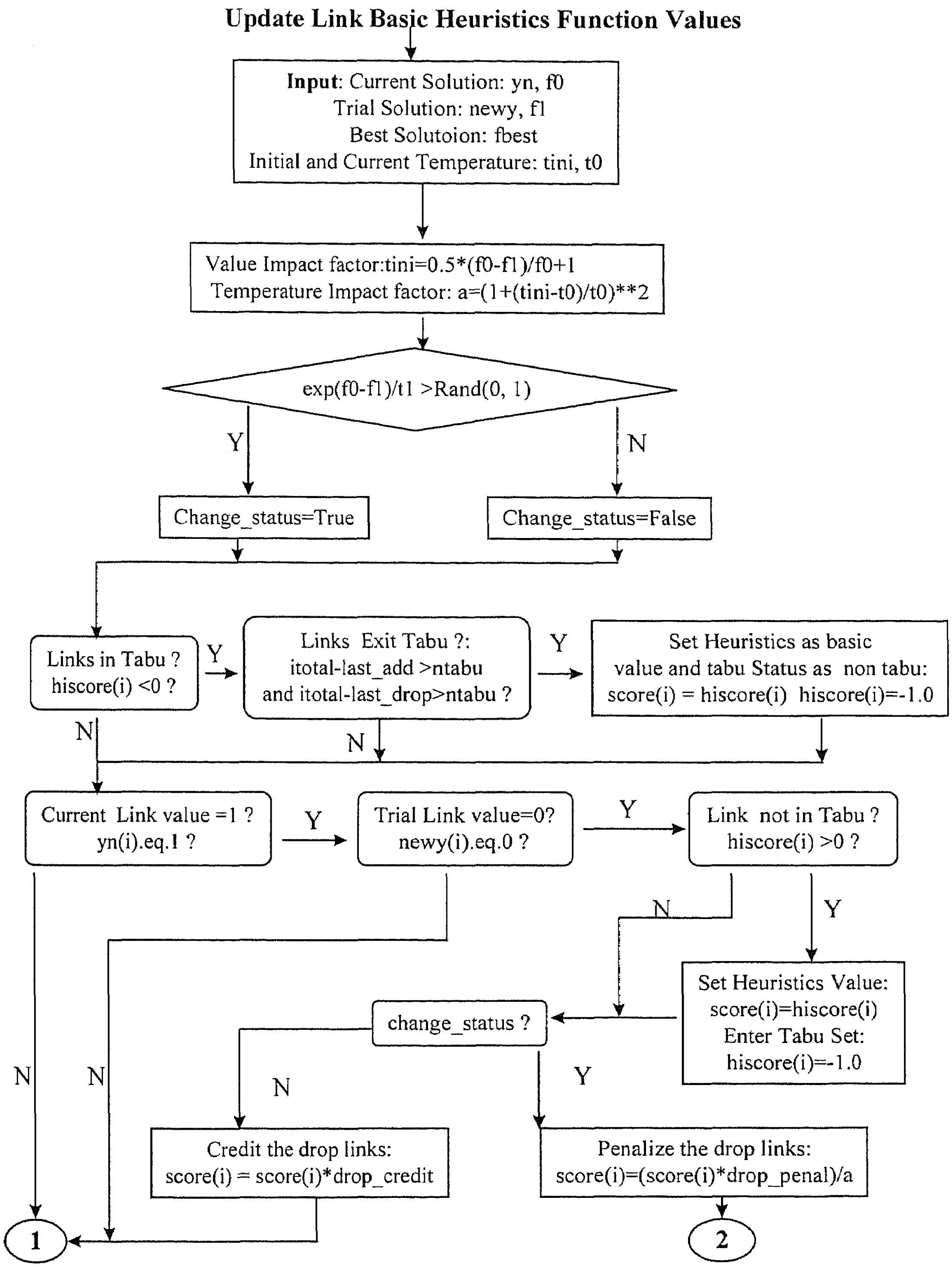


Figure 5.4.3 Link Basic Heuristics Value Updating Flow Chart (1/2)

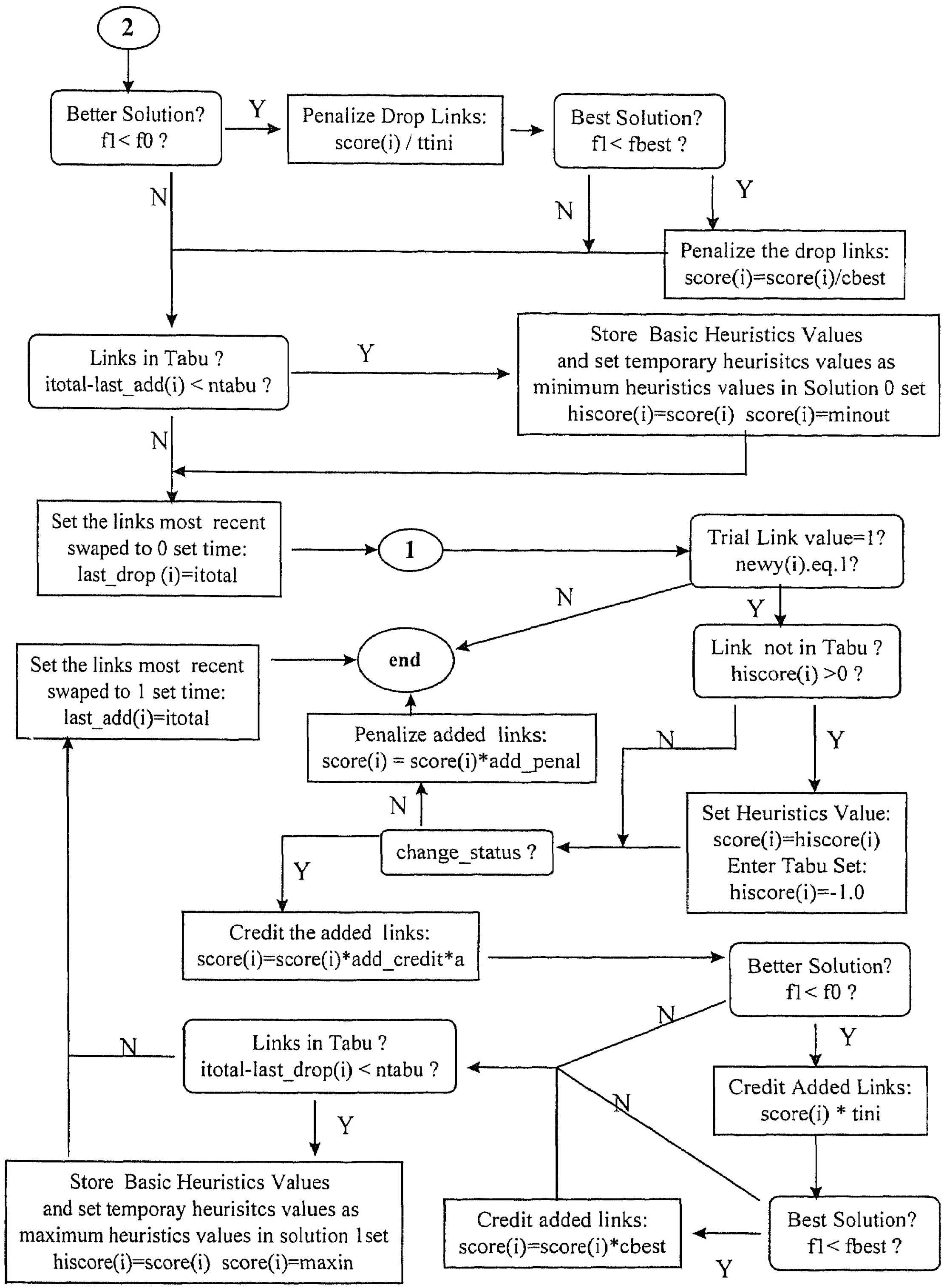


Figure 5.4.4 Link Basic Heuristics Value Updating Flow Chart (2/2)

For the links which are added to the Solution 1 set from the Solution 0 set, the link's LCOF value is updated in the opposite way to the procedure described for the dropped links in the preceding paragraph. If the move is accepted, the added links LCOF values are increased by the add_credit and value impact (a) factors. If the trial solution state has better value than the current solution state, the added link LCOF value is increased more by the "temperature" impact factor ttini. If the trial solution state is the best solution state, the LCOF value is factored more by the best solution factor (cbest). The added links would enter the tabu list by letting hiscore equal to score, score equal to the maximum HEF value of all the links of the Solution 1 set and last_add is set to the current iteration number.

After updating the link HEF values, the main program continues its process. If the trial solution state is accepted, the trial solution state would replace the current solution state ((f0, yn)=(f1, newy)), and the counter of the number of the straight trials that no trial solution state is accepted (no_change) is set to 0, while the counter of the number of the trial solution states accepted under the same "temperature" (yes_change) is incremented by 1, otherwise, the current solution state keeps unchanged and no_change counter is incremented by 1.

If the number of the transition (yes_change) reaches the current Markov chain length, the "temperature" will be decreased at a constant rate of 0.8 and simultaneously the Markov chain length (lm_length) is multiplied by 1.2 and the yes_change is set to 0 . The maximum Markov chain length is set to 21.

The termination of the program is determined by two factors. Either the number of the iteration reaches a preset maximum number of iterations or no acceptable trial solution states have been created for the last 25 trials (no_change = 25).

5.4.2 Subroutines

The main subroutines for this program involve procedures to generate the trial solution state. In the first step of the main program, the ***tperbation1*** subroutine is used to generate the trial solution state, that uses the random swapping strategy. In the second step of the main program, the ***tgrade_perbation*** subroutine is used to create trial solution states. Next, the subroutines ***tperbation1*** and ***tgrade_perbation1,*** as well as their supporting subroutines t***add_drops***, ***tadd_more*** and ***tdrop_more*** are described.

Subroutine *tperbation1* (Figure 5.4.5-5.4.6)

Subroutine ***tperbation1*** randomly swaps the links between the Solution-1 set and the Solution-0 set.

The subroutine's input data include the current solution state Solution-1 set, the Solution-0 set, the budget, the current solution state total budget cost, the link cost and the number of the passenger car links (n).

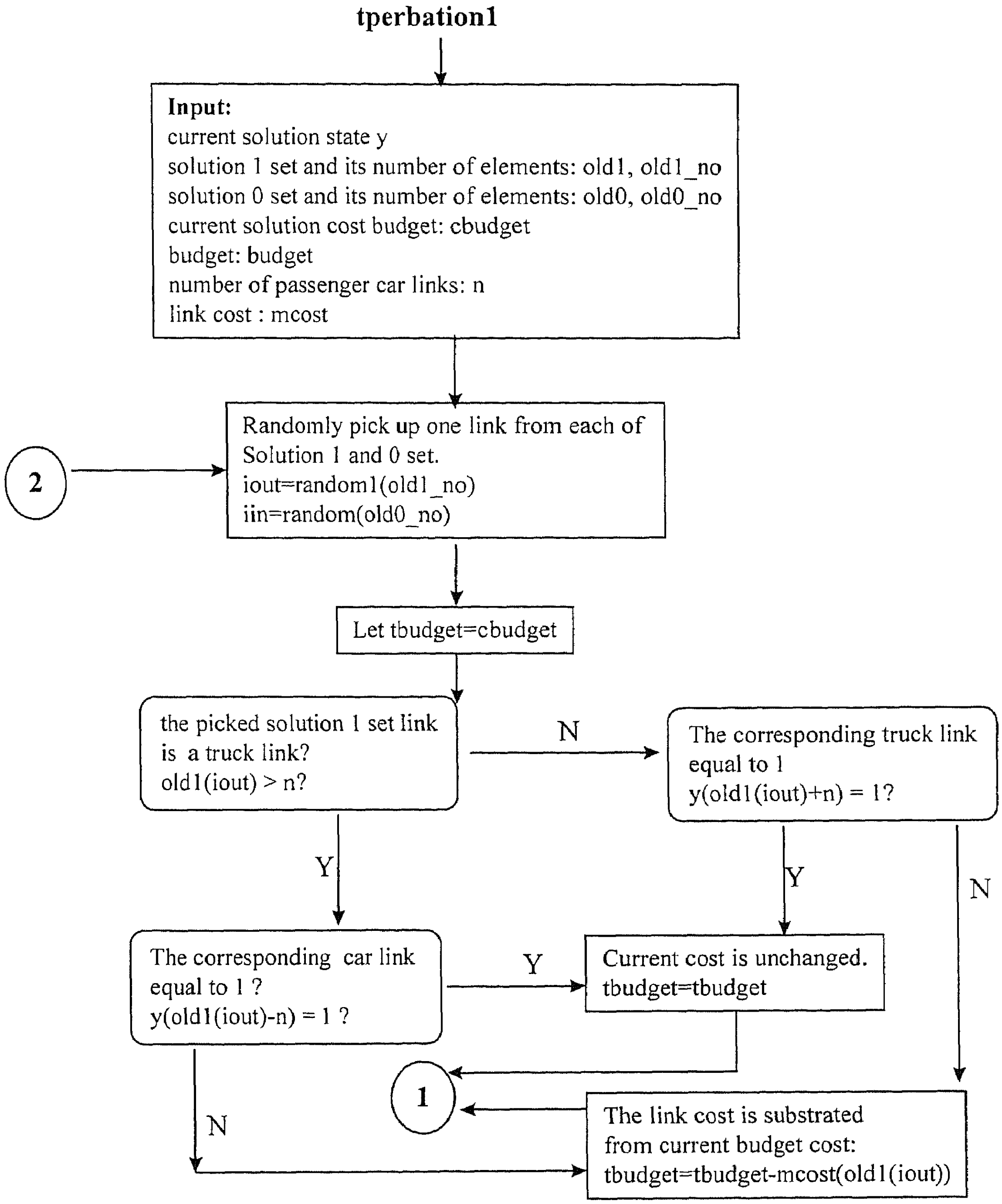


Figure 5.4.5 Subroutine *tperbation1* Flow Chart (1/2)

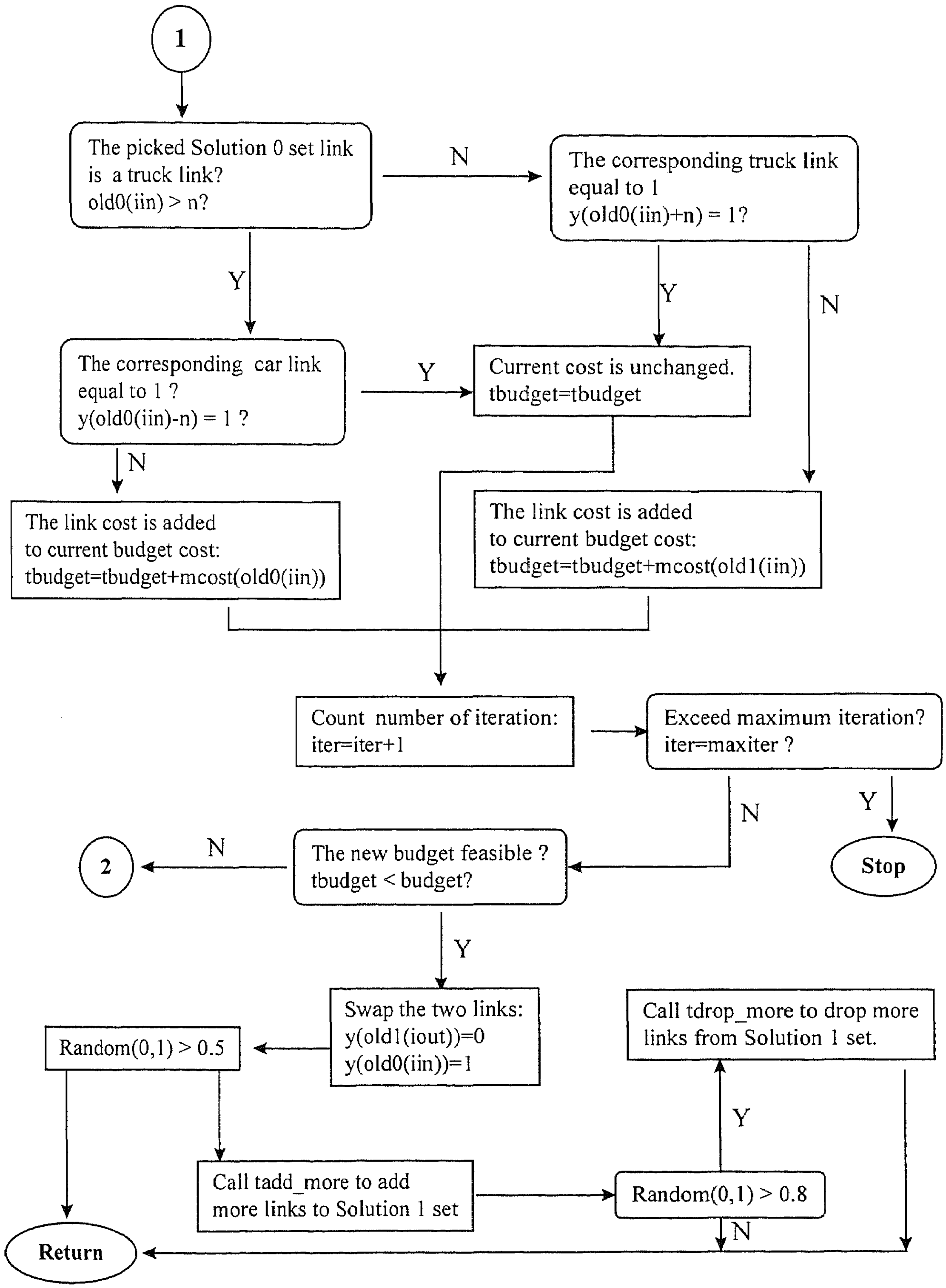


Figure 5.4.6 Subroutine *tperbation1* Flow Chart (2/2)

The subroutine's basic procedure is to randomly pick the links for swapping and then test if the new solution state violates the budget constraint. A random number is generated between 0 and 1 to the number of elements of the Solution-1 set (iout), and a random number for the Solution 0 set (iin) is also generated. Each of them represents the ith element in their sets. In general, the new budget cost, that is equal to the current budget cost minus the cost of ioutth link in Solution-1 set plus the cost of the iinth element in the Solution-0 set, is tested against the budget constraint (budget).

Whether the cost of the selected link to be dropped from the Solution 1 set to the Solution 0 set is subtracted from the current budget cost, depends, upon the status of the selected link's corresponding car or truck link. The program adds or subtracts the appropriate link cost based on the selected options. The new solution state will be accepted if the current budget cost is feasible, otherwise the same process is repeated by randomly picking another pair of the Solution 1 set and Solution 0 set to be swapped. After a maximum number of iterations is reached and a new feasible solution state is not found, the program will be terminated.

Subroutine ***tadd_more*** is called if a 0-1 random number is greater than 0.5. The ***tadd_more*** subroutine would add as many links as possible from the Solution-0 set to the Solution-1 set within the budget constraint. Since ***tadd_more*** subroutine is not necessarily needed at all times, only an average of 50% percent of time it is called. Similarly, at an average of 10% of the time the subroutine ***tdrop_more*** is called to drop more links from the Solution 1 set to Solution 0 set. The reason of adopting ***tdrop_more*** is that when the Solution 0 set links have a high budget cost, a feasible swapping could not be easily reached and the ***tdrop_more*** subroutine helps to consider a large number of links.

Subroutine *tadd_more* (Figure 5.4.7)

The function of subroutine ***tadd_more*** was presented in subroutine ***tperbation1***. The input variables are the same as of ***tperbation1***.

One link is randomly picked up in Solution 0 set (iin=random(old0_no)). The selected link is tested to find if it is a truck link or a passenger car link (is old0(iin)>n true ?, or is it a truck link ?), and its corresponding link's status is also checked (is y(old0(iin))n=1, true ? or is y(old(iin)-n)=1 true?). If the corresponding link is in the Solution-1 set, the current budget cost remains the same and the selected link is added to the Solution-1 set, otherwise the current budget will be increased by the selected link cost and the feasibility of the new current budget is checked (is tbudet < budget, true?). If the budget is not feasible, a new randomly picked link from Solution-0 set will be created and the preceding action will be repeated. This process will continue for a number of iterations until a feasible solution is found.

The addition of the selected link to the Solution 1 set requires the following steps: i) the selected link's value is set to 1; ii) the Solution 1 set number of elements is increased by one and the Solution-0 set is decreased by 1; iii) the budget cost is equal to the old budget cost plus the selected link cost; iv) the selected link's id is entered to the end of Solution-1 set's array and the elements in Solution-0 set are moved forward one position.

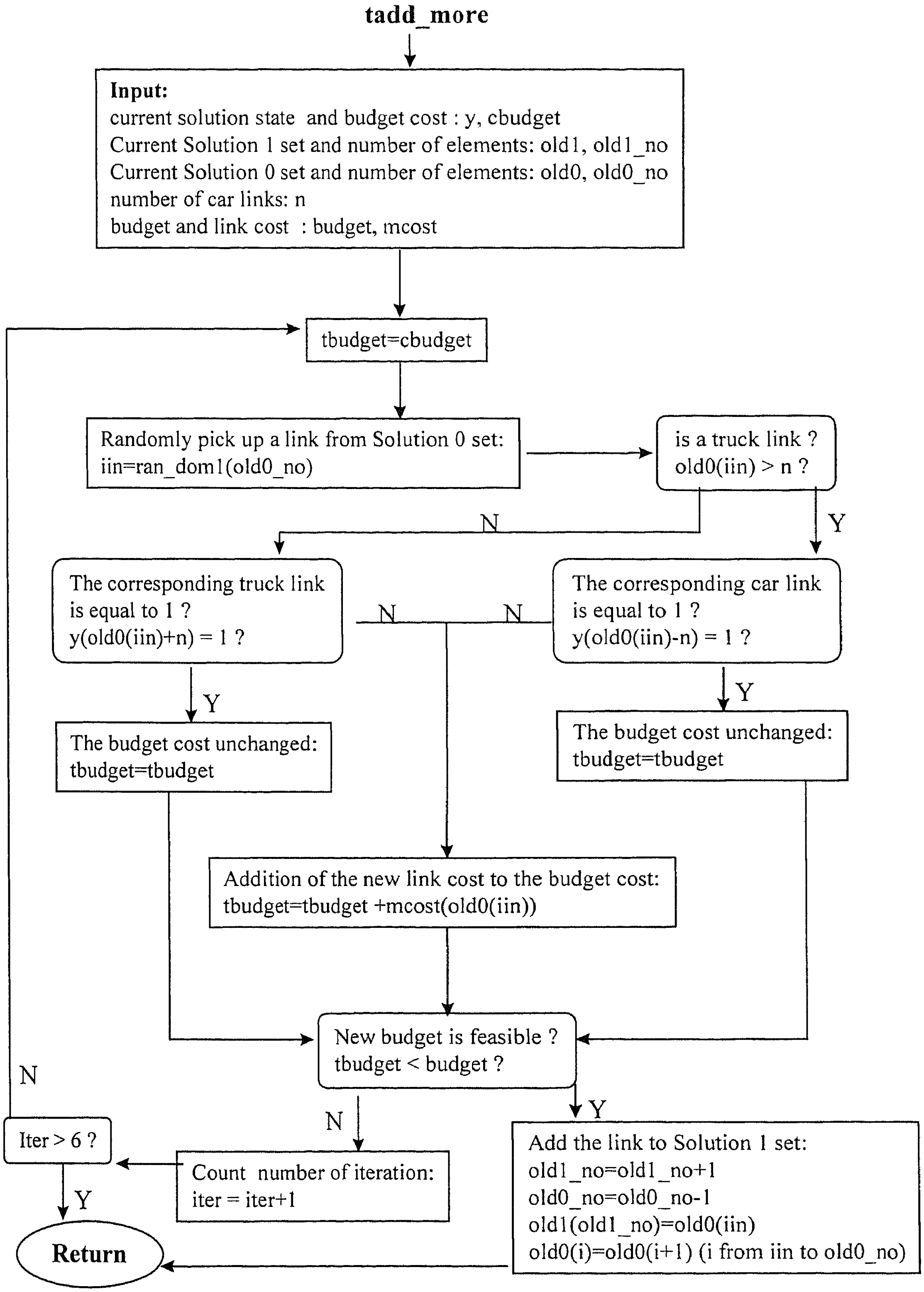


Figure 5.4.7 Subroutine *tadd_more* Flow Chart

Subroutine *tdrop_more* (Figure 5.4.8)

This subroutine is very similar to subroutine ***tdrop_more*** . It randomly picks up one link from Solution 1 set. The current budget is tested in a similar way as subroutine ***tdrop_more***. If the budget is changed, the task is completed, otherwise the process repeats until the budget changed or the number of iterations reaches 6. Other information can be referred to subroutine ***tdrop_more***.

Subroutine *tgrade_perbation1* (Figure 5.4.9)

This subroutine is used to generate the trial solution state. The primary required input variables include the current solution state (cy), link LCOF values (score), and the link's tabu status (hiscore).

The maximum and minimum LCOF values should be found first. The LCOF values (score) of all links except the links in the tabu list are scanned. The obtained maximum and minimum LCOF value are used in the next iteration.

The link's HEF values are those that are actually used. The LCOF values (score) present a summary information of the link's performance in the past trials of solving network design problem. The HEF (act_score) gathers the link congestion index information, and the LCOF values with random error to provide a more instructive stochastic information. The HEF value may vary with different designs of the search strategy. After getting the HEF values, the subroutine ***tadd_drops1*** is called to conduct the add/drop move based on the HEF values (act_score).

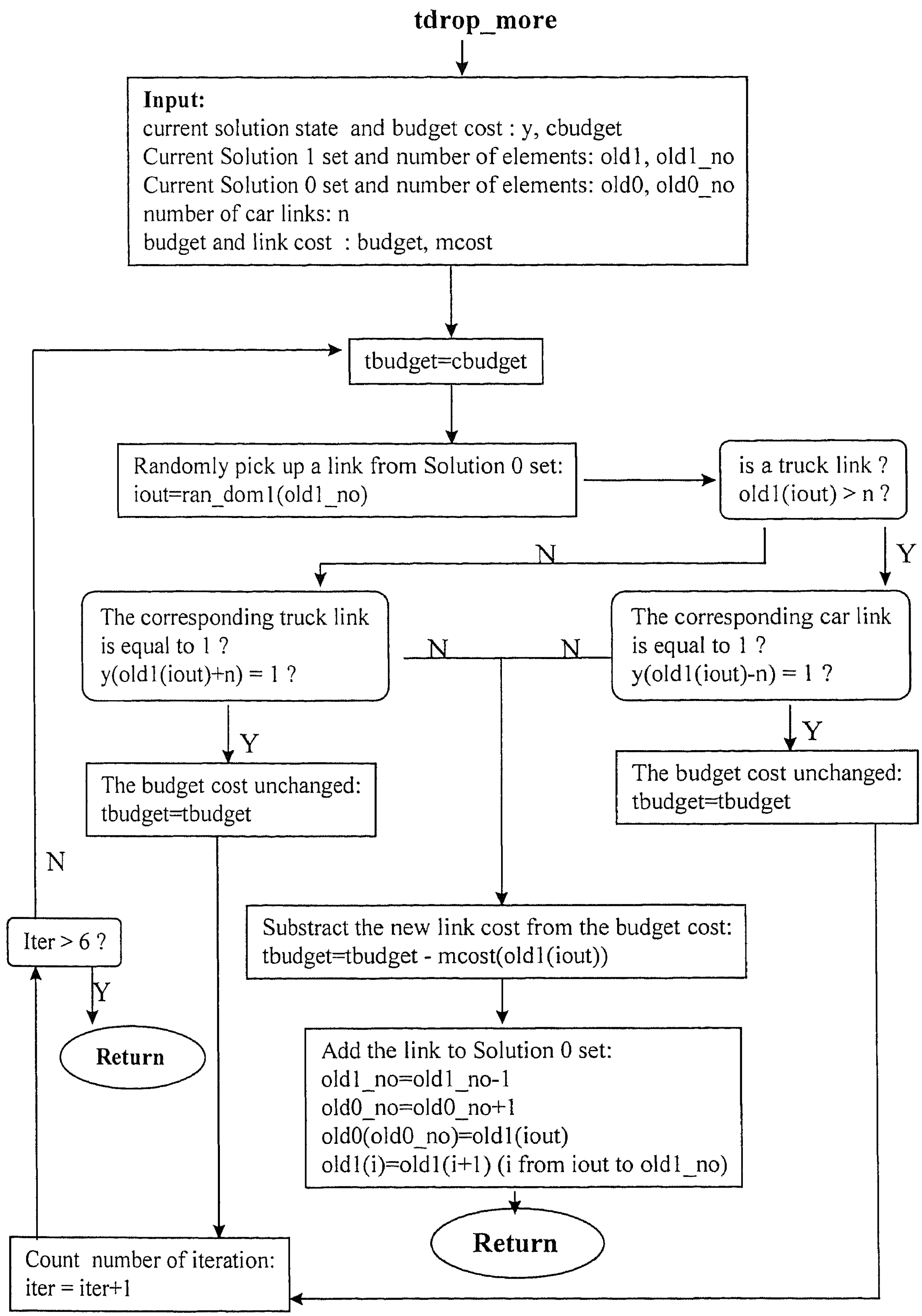


Figure 5.4.8 Subroutine tdrop_more Flow Chart

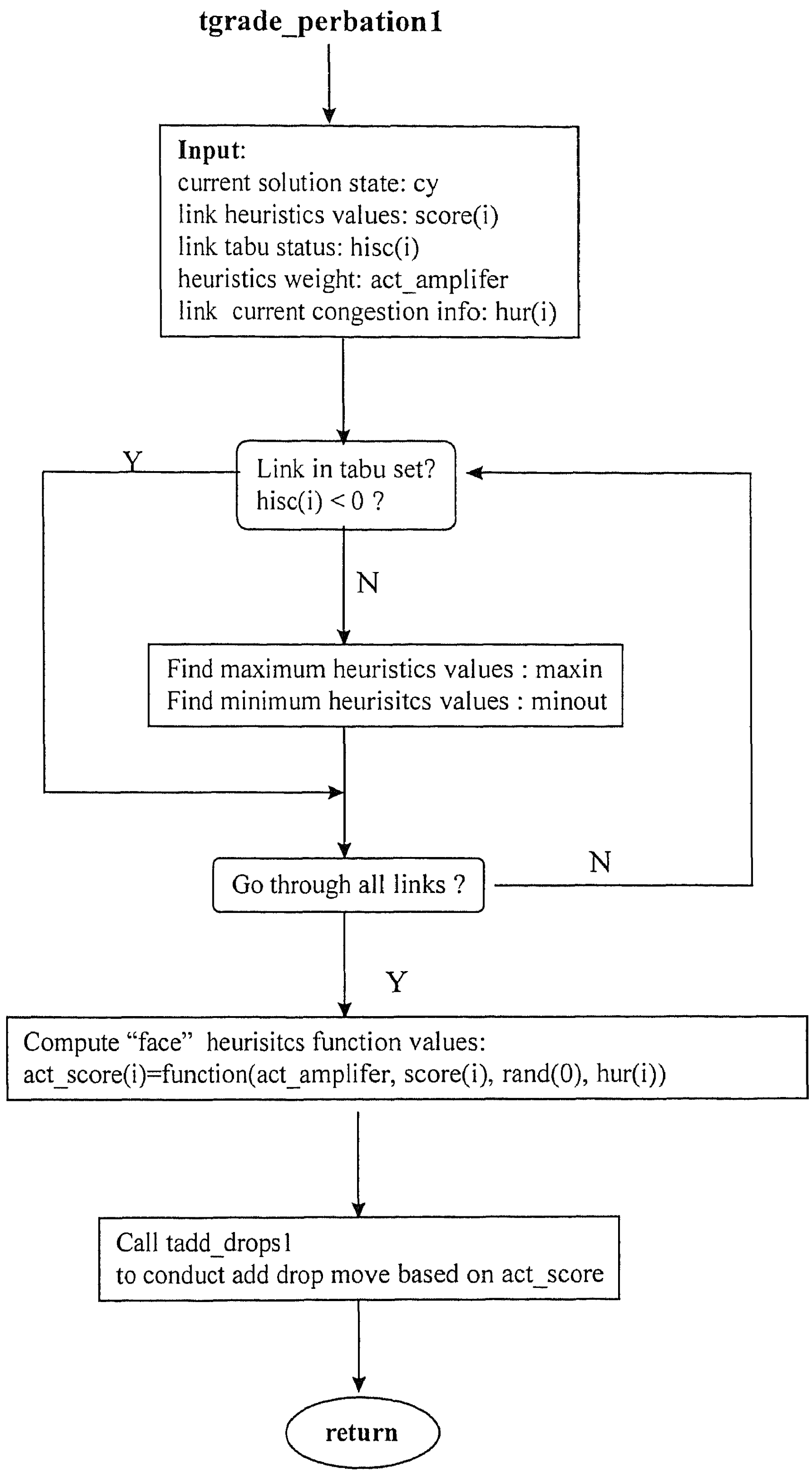


Figure 5.4.9 Subroutine *tgrade_perbation1* Flow Chart

Subroutine *tadd_drops1* (Figure 5.4.10 , 5.4.11, 5.4.12)

This subroutine performs the add/drop move based on the link's HEF values.

The important input data required is the current solution state (y), the current total cost (cbudget), number of car links (n), link cost (mcost), budget (budget), and link HEF values (heur).

The sorting subroutine, hpsort, is called to sort the Solution-1 set and Solution-0 set to orxin and orxout in decreasing order of the link HEF values (heur). The pointer for Solution-1 set is set to the first link of sorted Solution-1 set (Ismall=1) and for Solution-0 set, it is set to the last link of sorted Solution-0 set (itryl=k, k is the number of the elements in Solution-0 set).

Similar to subroutine ***tperbation1***, the status of the current Solution-0 set corresponding link is checked to determine whether the current budget cost remains the same or be updated by adding the selected link cost, if this link is added to Solution-1 set. If the current budget cost is not feasible, the Solution-0 set pointer keeps on moving to the next available link and the same action is conducted until all the available links in Solution 0 set are checked. Budget feasibility is continuously checked and more links become members of Solution-0 set until feasibility is reached.

The initial value for kadd and klast is set to zero. Variable kadd is incremented by 1 if the new feasible solution involves changing the exclusive lane to both car and truck operation lane. Variable klast is set to be equal to kadd after it is decided whether or not the new feasible solution should be accepted. The criteria for acceptance is

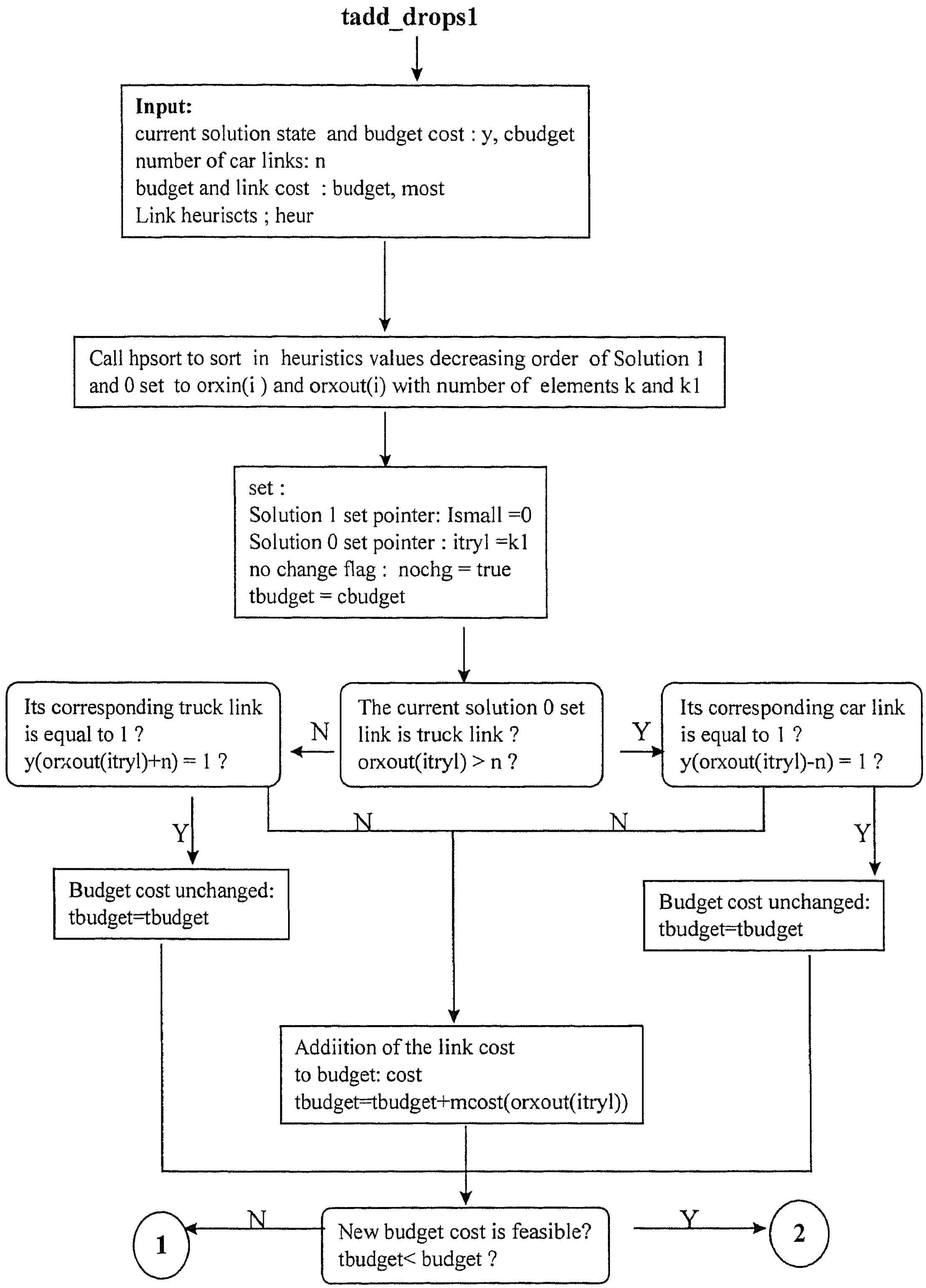


Figure 5.4.10 Subroutine *tadd_drops1* Flow Chart (1/3)

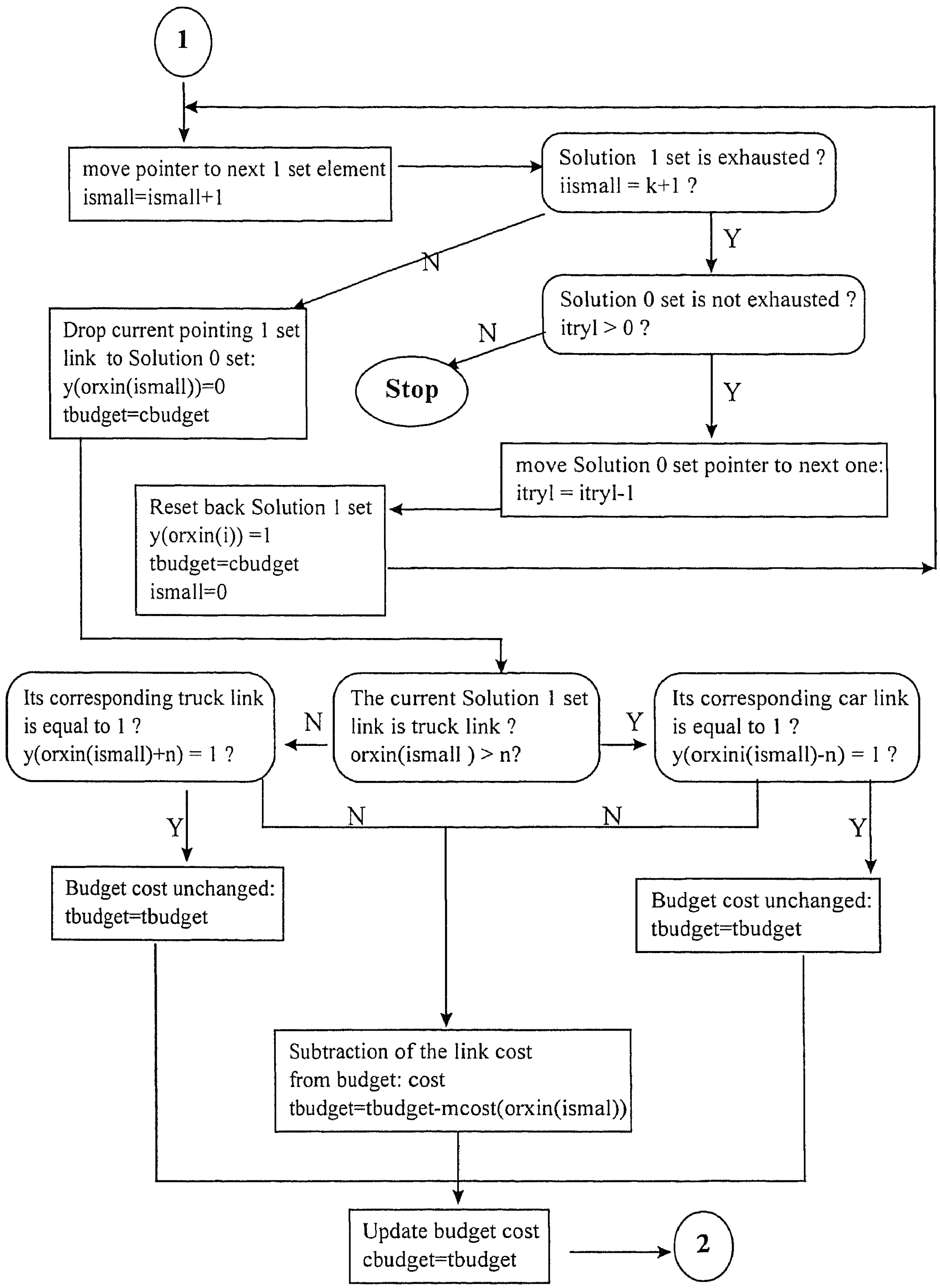


Figure 5.4.11 Subroutine *tadd_drops1* Flow Chart (2/3)

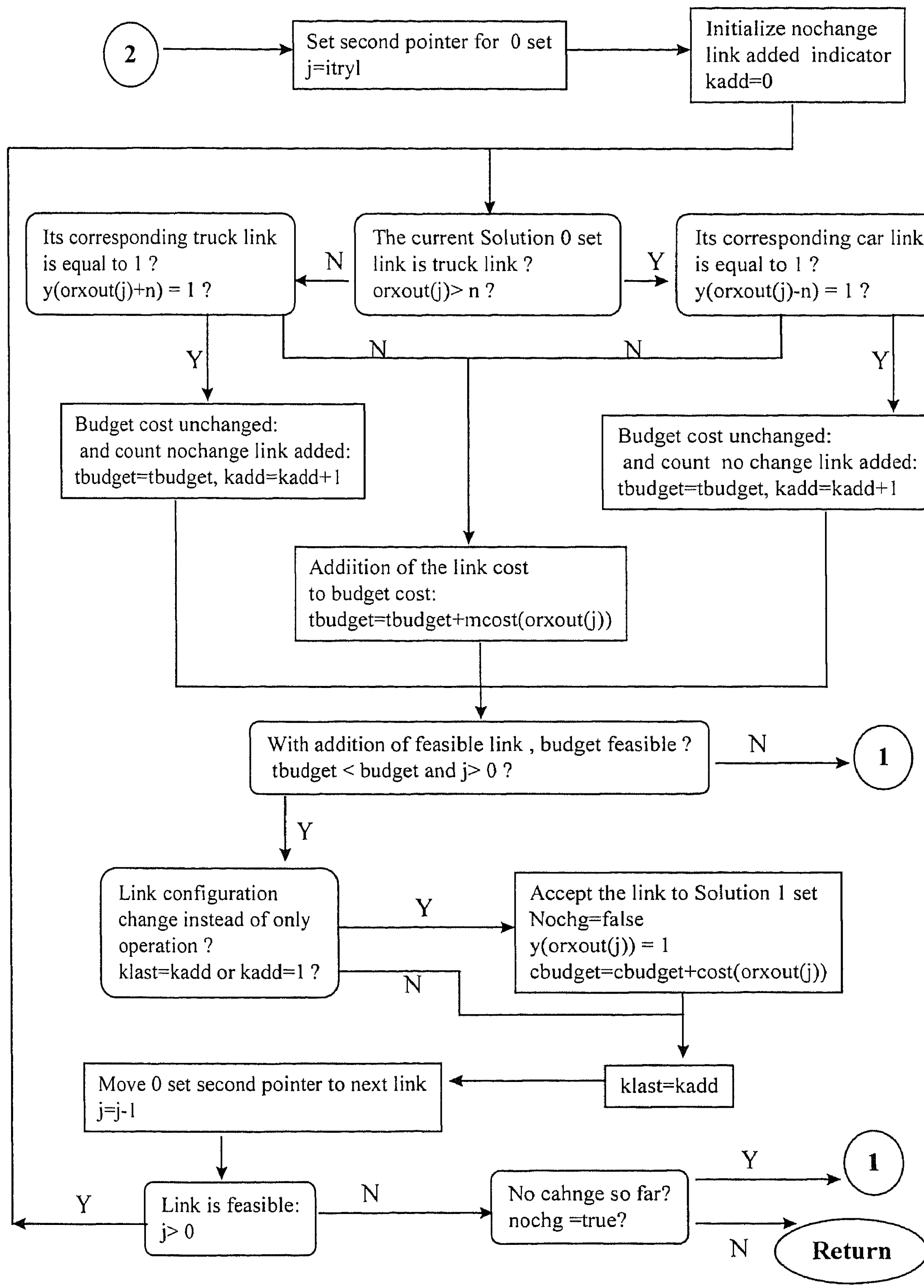


Figure 5.4.12 Subroutine *tadd_drops1* Flow Chart (3/3)

whether klast equal to kadd or kadd=1 is true. The first feasible solution will be automatically accepted no matter what kind of change is made. If the new feasible solution involves a change of the current budget cost, both kadd and klast are set to 0, and if a new feasible solution involves an non budget cost change (eg. changing from exclusive lanes to both passenger car and truck operation lane), kadd will be set to 1. After testing of the first feasible solution, the variable klast is set to the value of kadd. If the next feasible solution involves a change to the current budget changed, both variables kadd and klast remain the same and the condition kadd=klast is true, then the new feasible solution will be accepted as a new trial solution state. If the next feasible solution only involves a change of traffic operations in the link, the variable kadd would be increased by 1 and neither the conditions kadd=klast nor kadd=1 would be satisfied. This new feasible solution will be rejected as a new trial solution state.

CHAPTER 6

NUMERICAL EXPERIMENTS OF THE SA-TABU SEARCH STRATEGY

This chapter presents numerical experiments of the application of the SA-TABU search strategy. Five networks (Figures 4.4.1-4.4.5), ranging from 36 links to 363 links with various levels of budget constraint were tested. The primary objective of these numerical experiments is to examine the efficiency and effectiveness of the proposed of the proposed search strategy, and secondary is the test of the sensitivity of the key parameters of the SA-TABU.

6.1 Overview of the Numerical Experiment

The SA-TABU is a heuristic search strategy, which does not guaranteed to find an optimal solution. Thus, the difference of the search strategy's obtained "best" solution from the optimal solution could be used as a measurement of the solution quality of this heuristic. However, as mentioned in the previous chapters, the computational complexity of the TCNDP is non-polynomial, and with the current computing facilities, even for very small network problems, the computing time for obtaining the optimal solution required by the most efficient branch and bound algorithm can be out of practical feasible range. In this numerical experiment, it has been out of reach to compute the optimal solution for the five test networks with the available computing facilities. Therefore, it is not appropriate to use a very small network as the test network, just for obtaining the optimal solution while undermining the main purpose of this study - designing an efficient procedure for large network applications. Consequently, compromising criteria were employed in this study. The network total travel time when links are expanded by an extra lane for both passenger car and truck operations was set as a pseudo "upper" bound of the solution, and the difference between each specific problem's (different budget level) best solution and the "upper" bound value was used as the criterion to determine the quality of the SA-TABU.

Conventionally, a bench mark problem is needed to be used for comparison with the new proposed algorithm. However, in the current literature, except for the branch and bound algorithm (Mahmassani, et. al. 1984), no other algorithms have ever been developed which address the TCNDP, and no other bench mark problems have ever been set. The experiment ideally needs to cover as many networks with different configurations and characteristics as possible in the event of no bench mark problems is available. However, practically, the test problems need to be designed based on a reasonable number of experiments and good coverage of various types of transportation networks. Thus five different transportation networks (Figures 4.4.1-4.4.5) were used as test networks, which were associated with 10%, 20%, and 30% of available budget, respectively. These five test networks represent a variety of characteristics of real transportation networks (The five test networks were presented in Chapter 4).

Since no other algorithms have been developed to solve the TCNDP problem, the conventional simulated annealing algorithm is used as the only reference algorithm for comparison.

Some important features of the SA-TABU had been documented in early experiments on the single class of user network design problem application (Zeng et. al. 1996). Due to the complexity and lack of the optimal solution for the two classes of users network design problem in the test problems, these features could not be explored for the TCNDP, though they have been conducted for the solution of the single class user equilibrium network design problem. However, the principal characteristics of the procedure developed for the single class network design problem (SCNDP) were considered to be good candidates for the solution of the TCNDP as well. In this chapter, some important results that were obtained for the SCNDP are also presented since their basic rationale was followed for this study as well.

The five test networks trip tables are presented in Section 6.2. The relevant experimental results from the application of the SA-TABU on the SCNDP are summarized in Section 6.3. The numerical experiments on the comparison of the standard version the SA-TABU with the conventional simulated annealing algorithm in regard to the search strategy's efficiency are presented in Section 6.4. Sensitivity analysis conducted on different versions of the search strategies derived from the standard version is presented in Section 6.5, where the experiments are conducted by changing some key parameter values.

6.2 Test Networks

The five test networks (Figures 4.4.1 -4.4.5) and trip tables with 10%, 20% and 30% budget level in Tables 6.2.1-6.2.10 are used as the test problem in this study. A detailed description of the test networks' characteristics was presented in Chapter 4.

Table 6.2.1 Network 1 Passenger Car Trip Table

Centroids	1	2	3	4	5
1	--	500	2000	2000	1000
2	500	--	500	1000	1500
3	2000	500		1000	1500
4	2000	1000	1000	--	500
5	1000	1500	1500	500	--

Table 6.2.2 Network 1 Truck Trip Table

Centroids	1	2	3	4	5
1	--	50	200	200	100
2	50	--	50	100	150
3	200	50		100	150
4	200	100	100	--	50
5	100	150	150	50	--

Table 6.2.3 Network 2 Passenger Car Trip Table

Centroid	1	2	3	4
1	--	1000	2000	1500
2	1000	--	500	1500
3	2000	500	--	2000
4	1500	1500	2000	--

Table 6.2.4 Network 2 Truck Trip Table

Centroid	1	2	3	4
1	--	70	250	100
2	90	--	20	350
3	90	150	--	100
4	250	350	20	--

Table 6.2.5 Network 3 Passenger Car Trip Table

Centroids	1	2	3	4	5
1	--	1500	2000	3000	1000
2	500	--	500	2000	300
3	2000	3500		4000	3500
4	800	1000	1000	--	500
5	1000	1500	1500	2500	--

Table 6.2.6 Network 3 Truck Trip Table

Centroids	1	2	3	4	5
1	--	90	130	300	50
2	20	--	120	200	20
3	100	200		700	150
4	80	80	100	--	10
5	80	130	100	550	--

Table 6.2.7 Network 4 Passenger Car Trip Table

Centroids	1	2	3	4	5
1	--	1500	2000	3000	1000
2	500	--	500	2000	300
3	2000	3500	--	4000	3500
4	800	1000	1000	--	500
5	1000	1500	1500	2500	--

Table 6.2.8 Network 4 Truck Trip Table

Centroids	1	2	3	4	5
1	--	150	200	300	100
2	50	--	50	200	30
3	200	350	--	400	350
4	80	100	100	--	50
5	100	150	150	250	--

Table 6.2.9 Network 5 Passenger Car Trip Table

Centroid	1	2	3	4	5	6
1	--	900	300	100	300	200
2	900	--	800	600	3500	700
3	300	800	--	200	400	100
4	100	600	200	--	1100	200
5	300	3500	400	1100	--	1000
6	200	700	100	200	1000	--

Table 6.2.10 Network 5 Truck Trip Table

Centroid	1	2	3	4	5	6
1	--	130	90	100	30	100
2	10	--	10	400	350	100
3	10	20	--	300	40	100
4	20	20	50	--	110	100
5	10	90	20	900	--	100
6	10	30	10	300	100	--

6.3 Numerical Results of the SA-TABU Search Strategy in Solving the SCNDP

The five test networks presented in Section 6.2 were subjected to the application of the SA-TABU search strategies in solving the SCNDP, taken into consideration of the passenger car trip tables only. A summary of the results for the five networks are presented in Tables 6.3.1-6.3.5. In each of Network 1 with 10% to 30% budget and

Network 2 with 10% budget problems, where the optimal solutions are known, the SA-TABU search strategy found the optimal solutions. The optimal solutions for these networks were obtained through complete enumeration of the feasible solutions. The SA-TABU outperformed the conventional simulated annealing algorithm with better solutions and in less number of iterations.

In Figure 6.3.1, the entire solution space that consists of all feasible solutions for the Network 1 with a 30% budget problem and the solution space explored by the SA-TABU search strategy are presented. The graph shows that the SA-TABU search strategy focuses in searching the neighborhood of the best solutions, and wastes less time in the neighborhood of poor solutions. It partially explains the reason that the SA-TABU is more efficient and effective in solving the SCNDP.

The graph shown in Figure 6.3.2 provides an indication of the performance of the HEF developed in this study. Under the SA-TABU search strategy, the greater a link's HEF value is, the higher the probability that the link is selected to be expanded, or enter the Solution-1 set. The link's HEF values continue to be updated at every iteration. It can be observed in Figure 6.3.2 that, with the progress of the iterations, the gap between the average HEF values of the optimum solution links in the Solution-1 set and links in the Solution-0 set is steadily built up, and consequently the links in the optimal solution's Solution-1 set would have more and more chances to be selected. Eventually, the gap is so significant that only the optimal solution's Solution 1 set links are selected in the Solution 1 set by the search procedure and finally the search strategy converges after a certain number of iterations. This implies that the HEF developed is capable of distinguishing the good candidate links, and enables the search strategy to converge to a local optimal solution, which in this example is the global optimal.

Figures 6.3.3-6.3.5 demonstrate the performance of the SA-TABU, the conventional simulated annealing and the tabu search strategies (Mouskos, 1991) for Network 3 with a 30% budget level problem. The performance is measured by the objective function value or network total travel time during each iteration. The tabu search strategy (Figure 6.3.5) quickly approaches a good solution state - small network total travel time, in the early iterations, while however the solution does not seem to improve significantly as iterations progress. The conventional simulated annealing approach (Figure 6.3.4) does improve the solution as the iteration progresses, but most of the iterations are conducted in a poor solution state neighborhood. The SA-TABU (Figure 6.3.3) inherits the advantages of both simulated annealing and tabu search strategies, progresses in a superior solution state neighborhood and gradually approaches towards a much better solution state in an effective and robust manner.

Table 6.3.1 Summary of Numerical Experiment on Network 1 SCNDP with SA-TABU and Conventional Simulated Annealing

NETWORK INFORMATION		Network 1				
	Number of Links: 18	Number of Nodes: 6		Number of O-D Pairs: 20		
	Initial Network UE Travel Time (Upper Bound):		8593 Vehicle-Hours		Maximum Improvement:	5960
	All Links Expanded Network SO Travel Time (Lower Bound):		2633 Vehicle-Hours			
	Optimum Solution:	10% Budget	6281 Vehicle-Hours	38.79% Improvement		
		20% Budget	5000 Vehicle-Hours	60.29% Improvement		
		30% Budget	4121 Vehicle-Hours	75.03% Improvement		

PERFORMANCE OF THE ALGORITHMS	10% Budget		20% Budget		30% Budget	
	SA-TABU	SA	SA-TABU	SA	SA-TABU	SA
Best Solution	6,281	6,281	5,000	5,000	4,121	4,421
% Improvement	38.8%	38.8%	60.3%	60.3%	75.0%	70.0%
Number of Iteration	584	1,000	403	573	1,000	1,000
Termination Type	Converge	Converge	Converge	Converge	Max. Iteration	Max. Iteration
Starting Solution	7,807	6,281	6,751	5,368	5,025	6,740
Final Solution	6,281	6,281	5,009	5,009	4,121	4,448
Current Solution State						
Mean	6,399	6,322	5,192	5,424	4,255	4,413
Standard Deviation	258	138	171	350	198	356
Current Trail Solution State						
Mean	6,753	6,559	5,341	5,669	4,430	4,710
Standard Deviation	446	282	243	383	264	375

Table6.3.2 Summary of Numerical Experiment on Network 2 SCNDP with SA-TABU and Conventional Simulated Annealing.

NETWORK INFORMATION — Network 2

Number of Links: 38 — Number of Nodes: 10 — Number of O-D Pairs: 12

Initial Network UE Travel Time (Upper Bound): 1731 Vehicle-Hours — Maximum Improvement: 413

All Links Expanded Network SO Travel Time (Lower Bound): 1318 Vehicle-Hours

Optimum Solution:
- 10% Budget: 1570 Vehicle-Hours, 38.98% Improvement
- 20% Budget: Unknown
- 30% Budget: Unknown

PERFORMANCE OF THE ALGORITHMS

	10% Budget		20% Budget		30% Budget	
	SA-TABU	SA	SA-TABU	SA	SA-TABU	SA
Best Solution	1,570	1,579	1,472	1,466	1,410	1,416
% Improvement	39.1%	36.9%	62.8%	64.1%	77.8%	76.3%
Number of Iteration	1,000	330	1,000	878	1,000	858
Termination Type	Converge	Converge	Max. Iteration	Converge	Max. Iteration	Converge
Starting Solution	1,613	1,613	1,541	1,636	1,505	1,597
Final Solution	1,578	1,579	1,486	1,485	1,410	1,430
Current Solution State						
Mean	1,584	1,613	1,495	1,522	1,430	1,478
Standard Deviation	12	25	15	36	20	40
Current Trail Solution State						
Mean	1,603	1,629	1,509	1,542	1,441	1,492
Standard Deviation	16	25	17	33	19	36

Table6.3.3 Summary of Numerical Experiment on Network 3 SCNDP with SA-TABU and Conventional Simulated Annealing.

NETWORK INFORMATION		Network 3				
Number of Links: 64		Number of Nodes: 21		Number of O-D Pairs: 20		
Initial Network UE Travel Time (Upper Bound):			40,383	Vehicle-Hours	Maximum Improvement:	20,633
All Links Expanded Network SO Travel Time:			19,750	Vehicle-Hours		
Optimum Solution:	10% Budget	Unknown				
	20% Budget	Unknown				
	30% Budget	Unknown				

PERFORMANCE OF THE ALGORITHMS	10% Budget		20% Budget		30% Budget	
	SA-TABU	SA	SA-TABU	SA	SA-TABU	SA
Best Solution	26,357	26,618	23,486	23,537	22,012	22,807
% Improvement	68.0%	66.7%	81.9%	81.7%	89.0%	85.2%
Number of Iteration	1,000	557	1,000	946	1,000	630
Termination Type	Max.Iteration	Converge	Max. Iteration	Converge	Max. Iteration	Converge
Starting Solution	33,901	34,539	32,002	32,357	29,034	36,560
Final Solution	26,357	26,618	24,010	23,543	22,774	22,807
Current Solution State						
Mean	27,467	31,797	24,239	25,118	22,973	28,064
Standard Deviation	2,149	2,366	635	2,556	418	4,509
Current Trail Solution State						
Mean	28,175	33,387	24,561	26,156	23,202	28,660
Standard Deviation	2,604	3,048	1,016	2,695	764	4,351

Table 6.3.4 Summary of Numerical Experiment on Network 4 SCNDP with SA-TABU and Conventional Simulated Annealing.

NETWORK INFORMATION — Network 4

Number of Links: 126	Number of Nodes: 33		Number of O-D Pairs: 110		
Initial Network UE Travel Time (Upper Bound):		10,794	Vehicle-Hours	Maximum Improvement:	4,374
All Links Expanded Network SO Travel Time:		6,420	Vehicle-Hours		
Optimum Solution:	10% Budget	Unknown			
	20% Budget	Unknown			
	30% Budget	Unknown			

PERFORMANCE OF THE ALGORITHMS

	10% Budget		20% Budget		30% Budget	
	SA-TABU	SA	SA-TABU	SA	SA-TABU	SA
Best Solution	7,408	7,444	6,820	7,181	6,677	6,988
% Improvement	77.4%	76.6%	90.8%	82.6%	94.1%	87.0%
Number of Iteration	548	802	1,000	874	572	1,000
Termination Type	Converge	Converge	Converge	Converge	Converge	Max. Iteration
Starting Solution	8,716	9,308	8,641	8,906	8,781	8,923
Final Solution	7,624	7,640	6,843	7,181	6,720	6,988
Current Solution State						
Mean	7,640	7,873	6,946	7,611	6,769	7,207
Standard Deviation	97	376	162	462	128	276
Current Trail Solution State						
Mean	7,733	8,024	6,990	7,685	6,794	7,272
Standard Deviation	179	384	160	465	108	278

Table 6.3.5 Summary of Numerical Experiment on Network 5 SCNDP with SA-TABU and Conventional Simulated Annealing

NETWORK INFORMATION					
	Network 5				
Number of Links: 166	Number of Nodes: 50		Number of O-D Pairs: 30		
Initial Network UE Travel Time (Upper Bound):		8,221	Vehicle-Hours	Maximum	4,544
All Links Expanded Network SO Travel Time:		3,678	Vehicle-Hours	Improvement:	
Optimum Solution: 10% Budget	Unknown				
20% Budget	Unknown				
30% Budget	Unknown				

PERFORMANCE OF THE ALGORITHMS	10% Budget		20% Budget		30% Budget	
	SA-TABU	SA	SA-TABU	SA	SA-TABU	SA
Best Solution	5,006	5,375	4,240	4,631	3,996	4,227
% Improvement	70.8%	62.7%	87.6%	79.0%	93.0%	87.9%
Number of Iteration	156	540	1,000	556	1,000	1,000
Termination Type	Converge	Converge	Max. Iteration	Converge	Max. Iteration	Max. Iteration
Starting Solution	7,432	7,400	6,571	6,283	6,300	6,416
Final Solution	5,006	5,375	4,287	4,648	4,060	4,241
Current Solution State						
Mean	5,701	6,301	4,354	5,243	4,097	4,703
Standard Deviation	526	619	213	579	209	463
Current Trail Solution State						
Mean	5,785	6,424	4,428	5,327	4,143	4,761
Standard Deviation	460	585	241	562	220	462

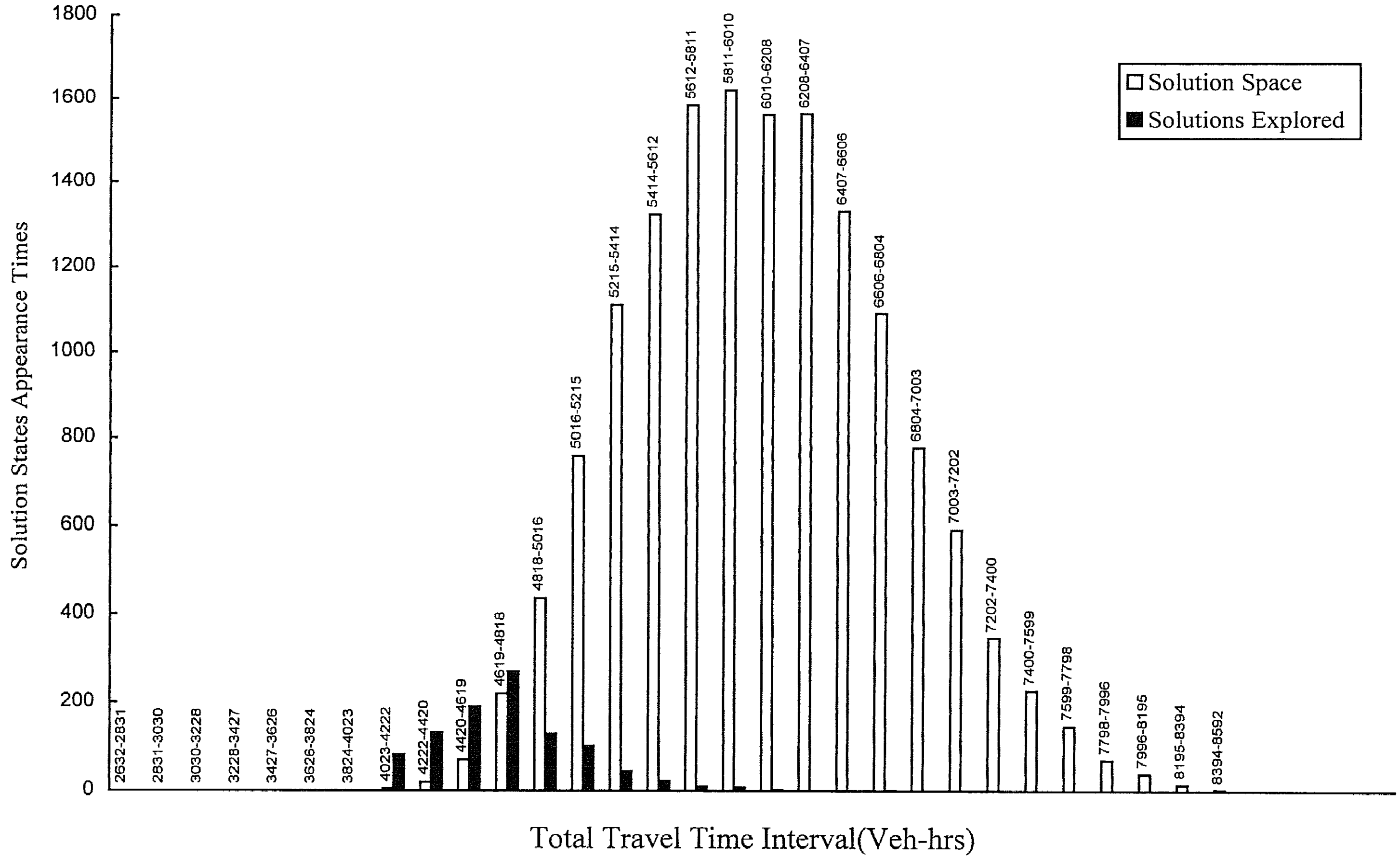


Figure 6.3.1 Frequency of the Solution States Visited By SA-TABU in Solving the SCNDP; (Network 1 30% Budget Level)

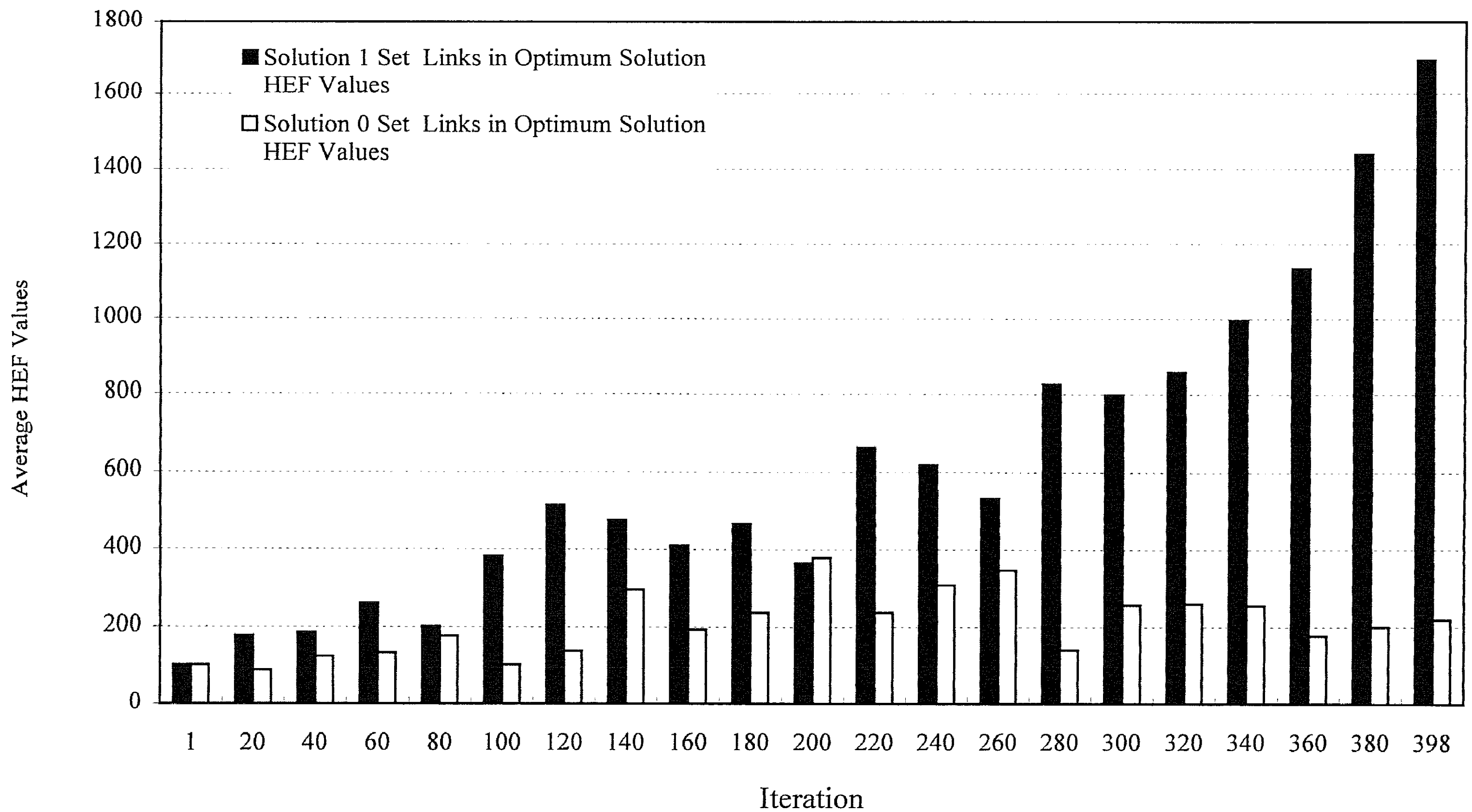


Figure 6.3.2 Average HEF Values of the Links in the Optimal Solution (Solution-1 Set) and the Links in the Solution-0 Set vs. Iteration; (Network 1, 10% Budget Level, SA-TABU in Solving SCNDP)

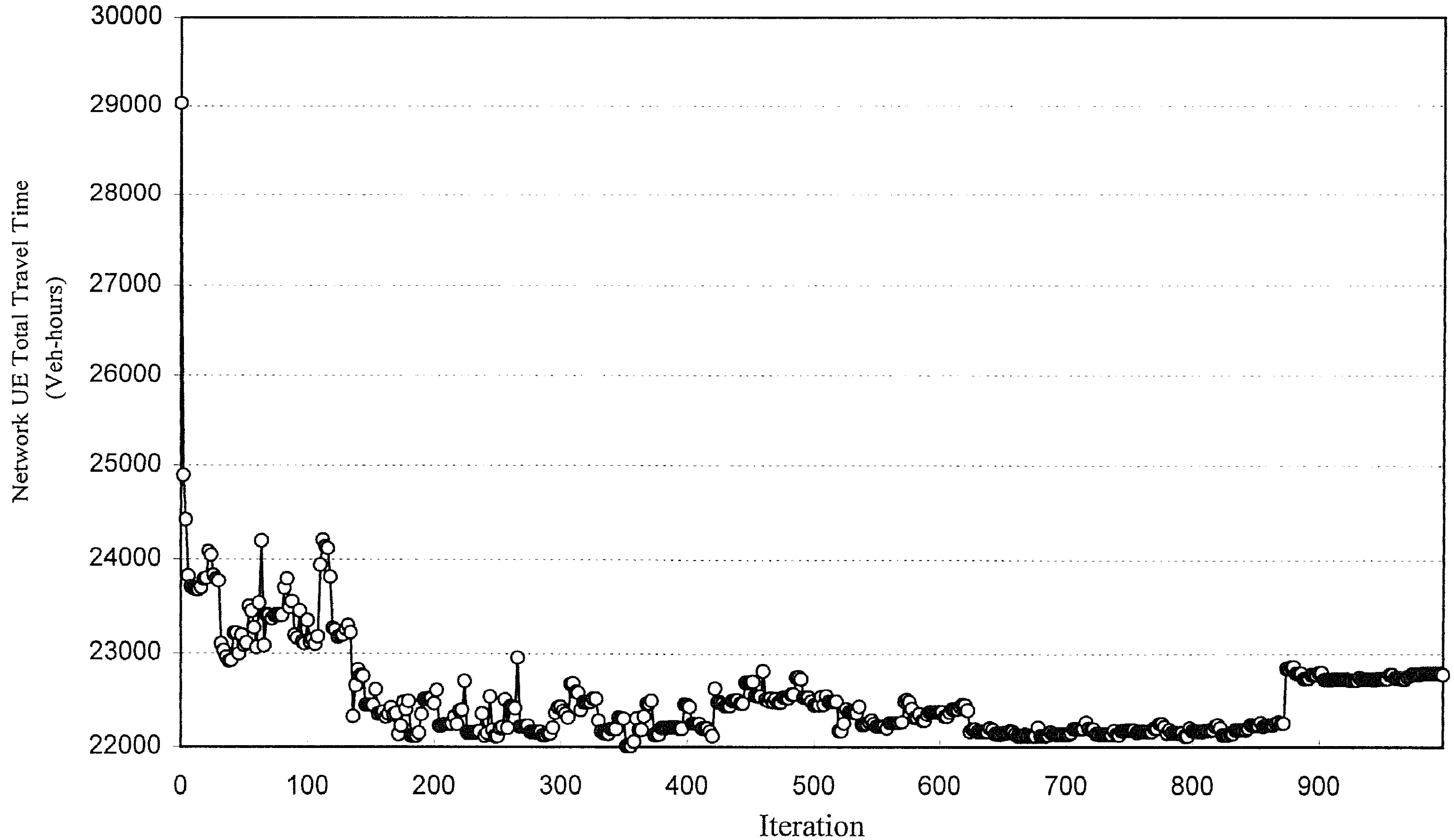


Figure 6.3.3 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, SA-TABU in Solving the SCNDP)

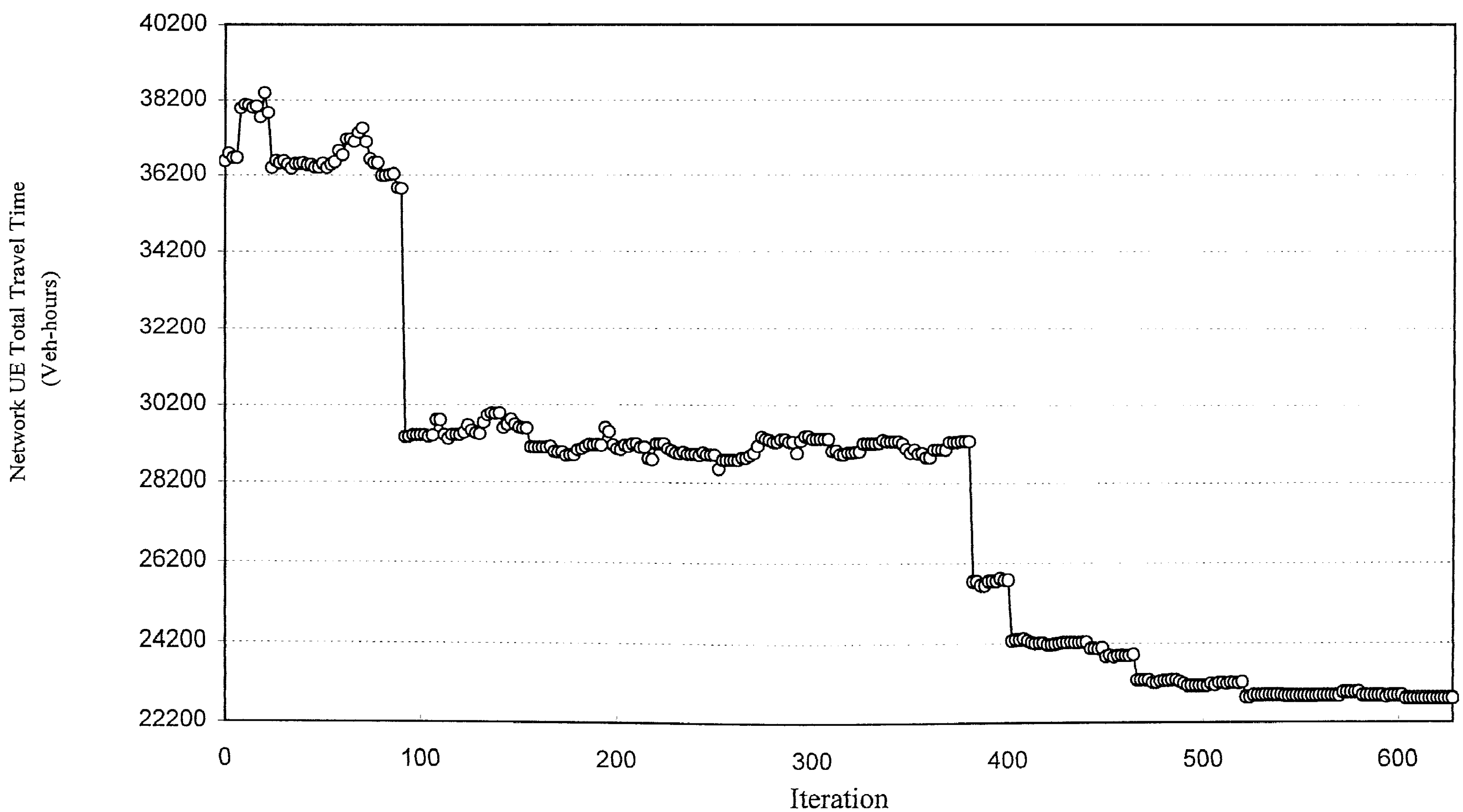


Figure 6.3.4 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, Simulated Annealing Algorithm in Solving the SCNDP)

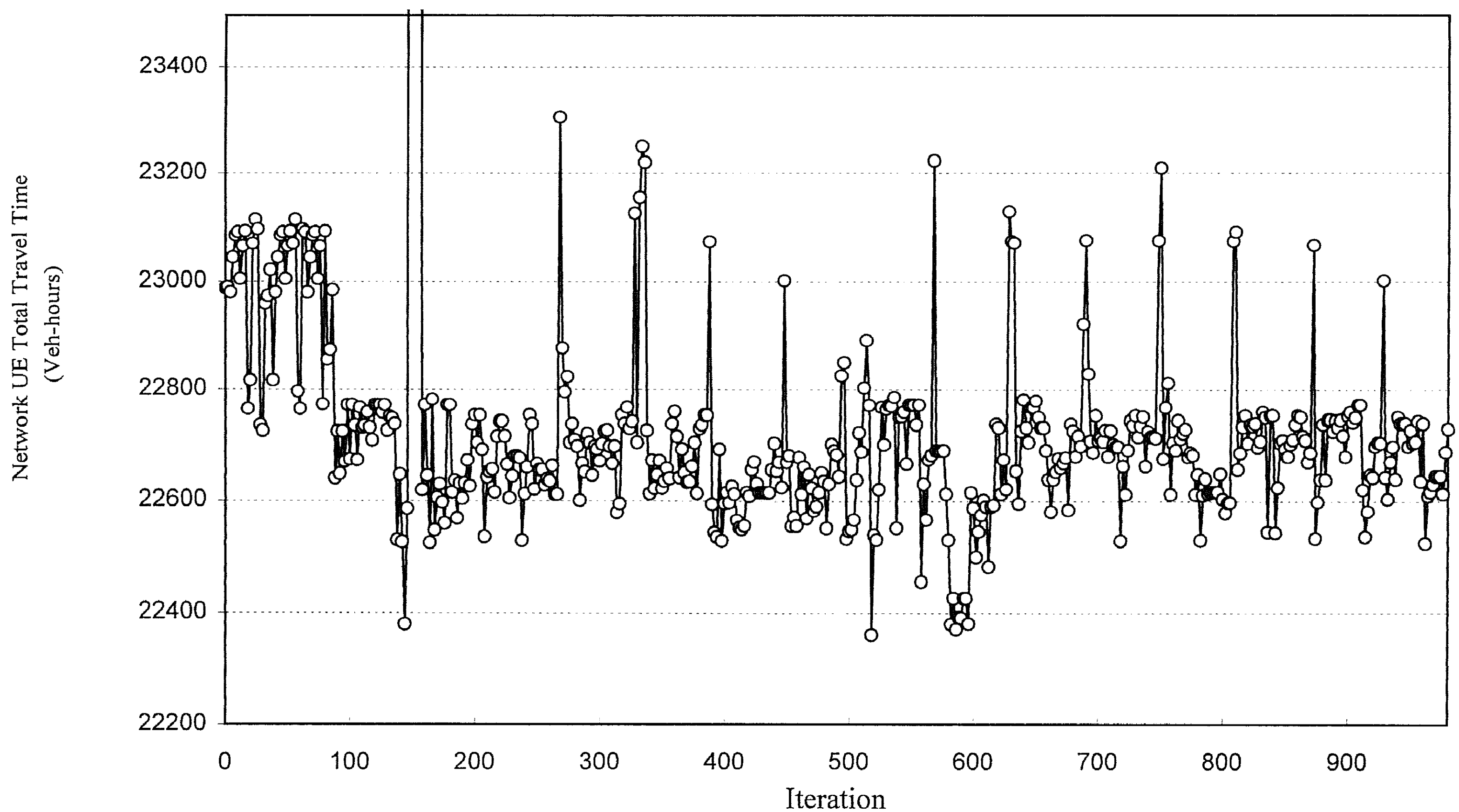


Figure 6.3.5 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, Tabu Search Strategy in Solving the SCNDP)

6.4 Numerical Experiments of the SA-TABU Search Strategy in Solving the TCNDP Problem

The primary difference between the SA-TABU strategy and the conventional simulated annealing algorithm is the methodology of generating the new trial solution state. In the conventional simulated annealing algorithm, the new trial solution state is generated by the random perturbation. However in the SA-TABU, a heuristics based add/drop type move, which incorporates a comprehensive HEF and tabu search characteristics, is used.

A number of different versions of the SA-TABU search strategies have been developed for the study. A standard version of the strategy is defined as a reference algorithm for comparison purposes in the sensitivity analysis study. The standard version has the following parameter settings: i) The control parameter ("temperature") decreasing rate is 85 %, while the Markov chain length is 21 and its increasing rate is 20%; ii) The tabu list length is equal to 50% of the total number of the network links; iii) The HEF random number factor is 0.5.

The conventional simulated annealing algorithm uses the same "temperature" and Markov chain length parameters which were used in order to have an unbiased comparison with the standard version of SA-TABU search strategy.

Tables 6.4.1-5 provide the statistical summary of the numerical experiments on the five test networks with 10%, 20% and 30% budget. From the summary tables, the following observations are made:

Table 6.4.1 Summary of the Numerical Results of the Application of the SA-TABU and SA; Network1;TCNDP

NETWORK 1 -------------- 36 Links, 12 Nodes

GENERAL INFORMATION

	Total	Passenger Cars	Trucks
Demand Trip Tables:			
Trips	25,300	23,000	2,300
Number of O-D Pairs	40	20	20
Total Network Travel Time:(Vehicle Hours)			
Initial Network:	14,455	13,499	956
Network with all links expanded for both pasenger cars and trucks:	3,739	3,495	244
Maximum Improvement:	10,716	10,004	712

PERFORMANCE OF THE ALGORITHMS

	10% Budget				20% Budget				30% Budget			
	SA	% Imp.	SA-TABU	% Imp.	SA	% Imp.	SA-TABU	% Imp.	SA	% Imp.	SA-TABU	% Imp.
Best Solution	7541	64.5%	6605	73.3%	7296	66.8%	6488	74.3%	6232	76.7%	6232	76.7%
Number of Iteration	1233		788		1560		1002		1823		310	
Termination Type	Converge		Converge		Converge		Converge		Converge		Converge	
Starting Solution	8753		8972		9678		7499		8812		6933	
Iteration Best Sol. First Found	980		656		1001		544		1756		103	
Current Solution State												
Mean	7934		6992		7651		6603		6433		6243	
Standard Deviation	203		114		192		168		177		89	
Current Trial Solution State												
Mean	8117		7117		7901		6765		6821		6269	
Standard Deviation	224		129		235		189		221		101	

Table 6.4.2 Summary of the Numerical Results of the Application of the SA-TABU and SA; Network2;TCNDP

NETWORK 2 ------------- 76 Links, 20 Nodes

GENERAL INFORMATION

	Total	Passenger Cars	Trucks
Demand Trip Tables:			
Trips	18,840	17,000	1,840
Number of O-D Pairs	24	12	12
Total Network Travel Time:(Vehicle Hours)			
Initial Network:	2,460	2,317	143
Network with all links expanded for both pasenger cars and trucks:	1,562	1,424	138
Maximum Improvement:	898	893	5

PERFORMANCE OF THE ALGORITHMS

	10% Budget				20% Budget				30% Budget			
	SA		SA-TABU		SA		SA-TABU		SA		SA-TABU	
		% Imp.		% Imp.		% Imp.		% Imp.		% Imp.		% Imp.
Best Solution	2232	25.4%	2171	32.2%	1876	65.0%	1720	82.4%	1765	77.4%	1693	85.4%
Number of Iteration	2000		812		2000		1208		2000		1630	
Termination Type	Max. Iteration		Converge		Max. Iteration		Converge		Max. Iteration		Converge	
Starting Solution	2354		2249		2309		2275		1939		1992	
Iteration Best Sol. First Found	1333		467		1578		1167		1602		1426	
Current Solution State												
Mean	2248		2242		2051		1920		1868		1766	
Standard Deviation	55		41		164		155		144		102	
Current Trial Solution State												
Mean	2255		2249		2173		2025		1897		1805	
Standard Deviation	63		48		207		186		193		184	

Table 6.4.3 Summary of the Numerical Results of the Application of the SA-TABU and SA; Network3;TCNDP

NETWORK 3 -------------- 128 Links, 42 Nodes

GENERAL INFORMATION

	Total	Passenger Cars	Trucks
Demand Trip Tables:			
Trips	36,810	33,600	3,210
Number of O-D Pairs	40	20	20
Total Network Travel Time:(Vehicle Hours)			
Initial Network:	72,496	67,144	5,352
Network with all links expanded for both pasenger cars and trucks:	36,618	34,143	2,475
Maximum Improvement:	35,878	33,001	2,877

PERFORMANCE OF THE ALGORITHMS

	10% Budget				20% Budget				30% Budget			
	SA		SA-TABU		SA		SA-TABU		SA		SA-TABU	
		% Imp.		% Imp.		% Imp.		% Imp.		% Imp.		% Imp.
Best Solution	40,213	90.0%	38,584	94.5%	37,000	98.9%	37,024	98.9%	37,001	98.9%	36,920	99.2%
Number of Iteration	2,000		1,536		2,000		1,028		2,000		1,756	
Termination Type	Max. Iteration		Converge		Max. Iteration		Converge		Max. Iteration		Converge	
Starting Solution	71,558		61,895		69,447		58,683		69,069		52,377	
Iteration Best Sol. First Found	1,899		1,164		1,772		846		1,926		1,225	
Current Solution State												
Mean	48,768		42,624		45,786		39,399		41,991		40,441	
Standard Deviation	8,619		1,183		4,546		1,024		5,440		2,669	
Current Trial Solution State												
Mean	52,171		44,878		46,372		41,460		42,104		41,509	
Standard Deviation	10,483		1,662		6,360		1,287		6,449		2,568	

Table 6.4.4 Summary of the Numerical Results of the Application of the SA-TABU and SA; Network4;TCNDP

NETWORK 4 -------------- 252 Links, 66 Nodes

GENERAL INFORMATION

	Total	Passenger Cars	Trucks
Demand Trip Tables:			
Trips	40,110	33,600	6,510
Number of O-D Pairs	40	20	20
Total Network Travel Time:(Vehicle Hours)			
Initial Network:	17,807	16,572	1,235
Network with all links expanded for both pasenger cars and trucks:	9,202	8,559	643
Maximum Improvement:	8,605	8,013	592

PERFORMANCE OF THE ALGORITHMS

	10% Budget				20% Budget				30% Budget			
	SA		SA-TABU		SA		SA-TABU		SA		SA-TABU	
		% Imp.		% Imp.		% Imp.		% Imp.		% Imp.		% Imp.
Best Solution	10,391	86.2%	10,520	84.7%	10,099	89.6%	9,756	93.6%	9,698	94.2%	9,343	98.4%
Number of Iteration	2,000		983		2,000		2,000		2,000		1,752	
Termination Type	Max. Iteration		Converge		Max. Iteration		Max. Iteration		Max. Iteration		Converge	
Starting Solution	17,550		16,476		17,062		15,337		16,029		14,057	
Iteration Best Sol. First Found	1,265		554		1,993		1,564		901		1,154	
Current Solution State												
Mean	11,455		11,320		11,028		10,278		10,175		10,171	
Standard Deviation	1,524		873		790		316		1,056		618	
Current Trial Solution State												
Mean	12,548		11,471		12,071		11,296		11,919		10,969	
Standard Deviation	1,941		859		1,099		693		1,969		774	

Table 6.4.5 Summary of the Numerical Results of the Application of the SA-TABU and SA; Network5;TCNDP

NETWORK 5 ------------- 332 Links, 100 Nodes

GENERAL INFORMATION

	Total	Passenger Cars	Trucks
Demand Trip Tables:			
Trips	24,460	20,800	3,660
Number of O-D Pairs	60	30	30
Total Network Travel Time:(Vehicle Hours)			
Initial Network:	246,141	220,927	25,214
Network with all links expanded for both pasenger cars and trucks:	16,604	10,695	5,909
Maximum Improvement:	229,537	210,232	19,305

PERFORMANCE OF THE ALGORITHMS

	10% Budget				20% Budget				30% Budget			
	SA		SA-TABU		SA		SA-TABU		SA		SA-TABU	
		% Imp.		% Imp.		% Imp.		% Imp.		% Imp.		% Imp.
Best Solution	24,483	96.6%	23,512	97.0%	22,019	97.6%	20,005	98.5%	19,973	98.5%	18,269	99.3%
Number of Iteration	2,000		2,000		2,000		2,000		2,000		2,000	
Termination Type	Max. Iteration		Max. Iteration		Max. Iteration		Max. Iteration		Max. Iteration		Max. Iteration	
Starting Solution	243,015		83,215		239,826		60,761		237,056		55,744	
Iteration Best Sol. First Found	1,820		738		1,537		1,656		1,997		1,764	
Current Solution State												
Mean	51,605		35,683		46,860		32,530		43,362		20,829	
Standard Deviation	46,739		9,836		30,793		6,913		59,481		7,706	
Current Trial Solution State												
Mean	53,805		39,762		49,499		39,623		48,757		28,936	
Standard Deviation	48,692		18,878		33,437		15,333		68,434		9,126	

1) The SA-TABU produced the better "best solution" in almost scenario, except for the Network 1 - 30% budget level and the Network 4 - 10% budget level problems. The advantage of the SA-TABU search strategies can be observed rather clearly for the problems with larger feasible solution spaces, they do manifest their edge over the conventional algorithms in almost every problem.

2) The conventional simulated annealing algorithm(SA) takes much more iterations than the SA-TABU to reach comparable solutions. Except for Network 1, it does not converge in all the problems (The term "converge" is redefined to mean that the search strategy stops because none of the trial solution states has been accepted in the last limited number of iterations). Except for large networks such as Network 5, the SA-TABU converged in most of the problems conducted within 2000 iterations. Under these criteria, the SA-TABU is fast and more effective. In small network problems such as Network 1, the SA converges in more iterations while it does not generate any better "best solution" than the SA_TABU search strategy. This may be explained by the fact that the "temperature" drops too fast and the algorithm falls into a local optimum. Though, it can not be guaranteed that the solution obtained by the SA-TABU search strategy is the optimum solution, it is very encouraging to observe that it can produce better "best solution" while the SA converges to a local optimum.

3) In the SA, the starting solution state is randomly generated, while the SA-TABU search strategy utilizes the heuristic's information to generate the starting solution state. The results provide a strong indication that the SA-TABU search strategy starts at a much more superior solution state. However, it is noted that in Network 1 with a

10% budget level and in Network 2 with a 30% budget level problems, the SA starts with a better starting solution state. As the network size and budget level increase, though the gap of the starting solution state quality between the SA-TABU and SA is widened. In the Network 5 - 30% problem, the difference is very large, which underscores one of the advantage of the SA-TABU over the SA. It is expected that for large scale problems, this advantage would be much more transparent.

4) The "best solution" appears in an earlier iteration for the SA-TABU search, compared with SA. As the SA-TABU searches a solution space which is formed based on heuristic information, the probability in finding the "best solution" is much larger than the SA which uses a more slow and conservative iterative process with no information. The comparison of the two procedures is not significant in small size Networks (e.g. 1 Network 1 and Network 2). However, as the network size increases, the SA finds its "best solution" close to the maximum iteration. That is an indication that the SA is too slow in reaching better solution for large scale networks.

5) The SA-TABU search strategy always has a smaller mean of the current solution state, and especially for the relatively large network - Network 5 problem, the difference of the current solution state mean between the two search strategies is significant. The mean of trial solution state value, showing the same property as the current solution state in the results, is the gathering of the generated searching space. The standard deviation of the current solution state values reflects the fluctuation of the current solution state. Similarly, the standard deviation of the trial solution state addresses the degree of the search in exploring the solution space. The tables

demonstrate that the SA explores a much wider searching solution space. Therefore, for small network applications, the SA may produce better solutions to the SA-TABU in some cases. However for large network applications, the SA-TABU strategies would be much more efficient and effective. Though most of the problem experiments support the above statement, the results from both Network 1 and Network 2 present a contradiction. This contradiction can be explained by the complexity of the algorithm and the nature of the network design problem, especially when a dynamic random variable is involved in the search process.

Figures 6.4.1 and 6.4.2 show the current solution state values at each iteration for the SA and the SA-TABU search strategies in Network 3-30% problem respectively. The trial solution state values per iteration for each of the two search strategies are depicted in Figures 6.4.3 and Figure 6.4.4 respectively. The observations from these figures are found to be consistent with the above discussion.

It may be concluded that the SA-TABU search strategies are much more efficient and effective than SA, especially for larger networks.

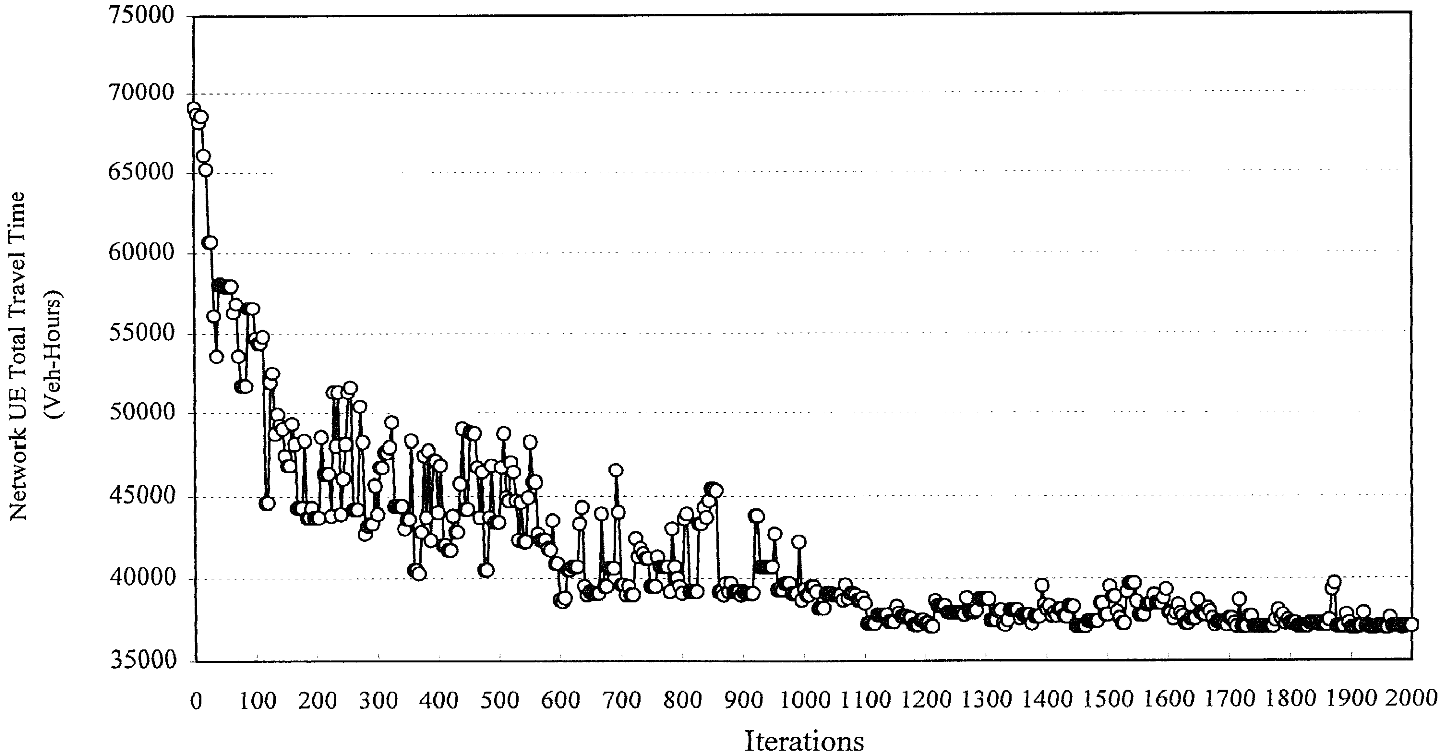


Figure 6.4.1 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, SA in Solving the TCNDP, Current Solution State)

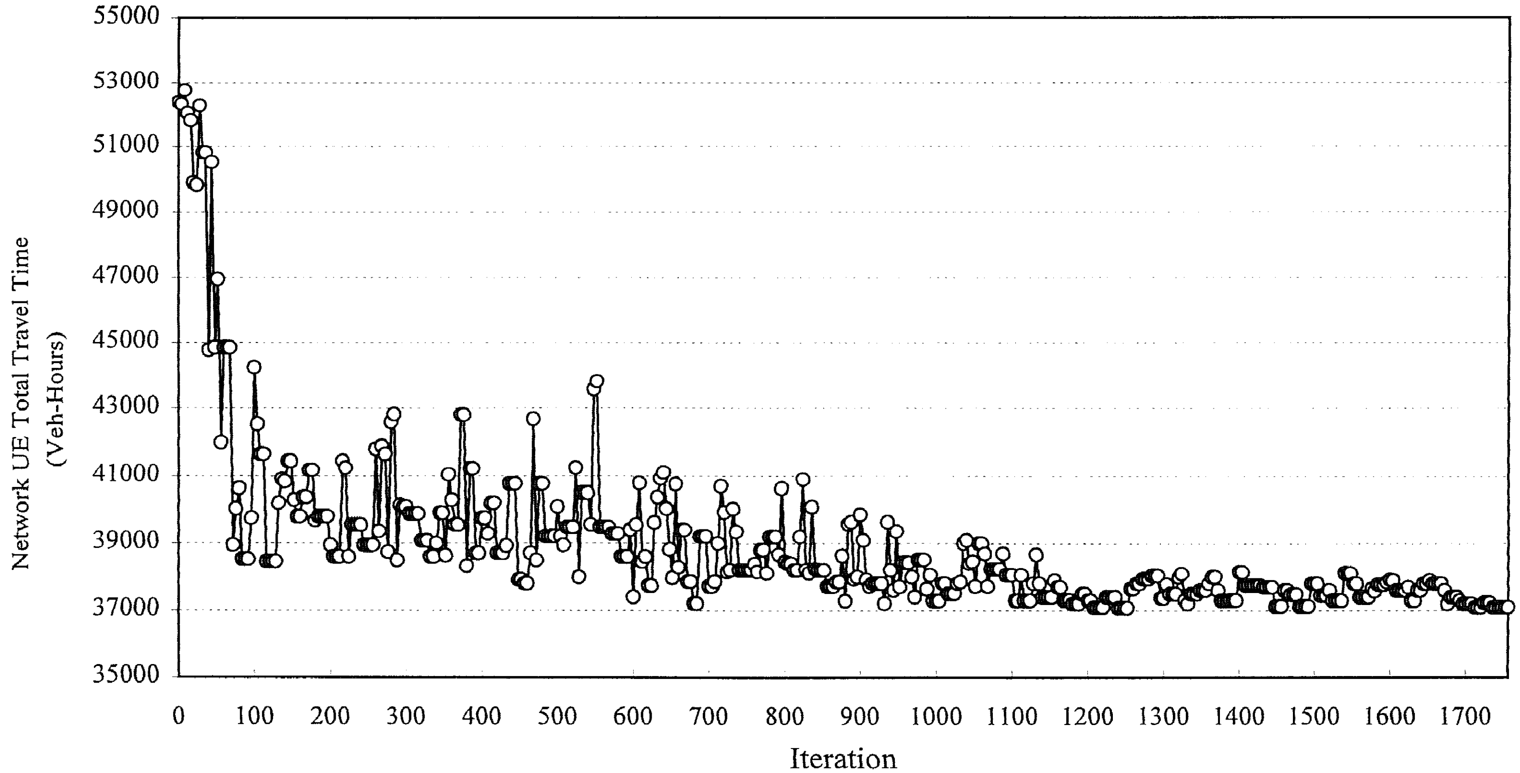


Figure 6.4.2 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, SA-TABU in Solving the TCNDP, Current Solution State)

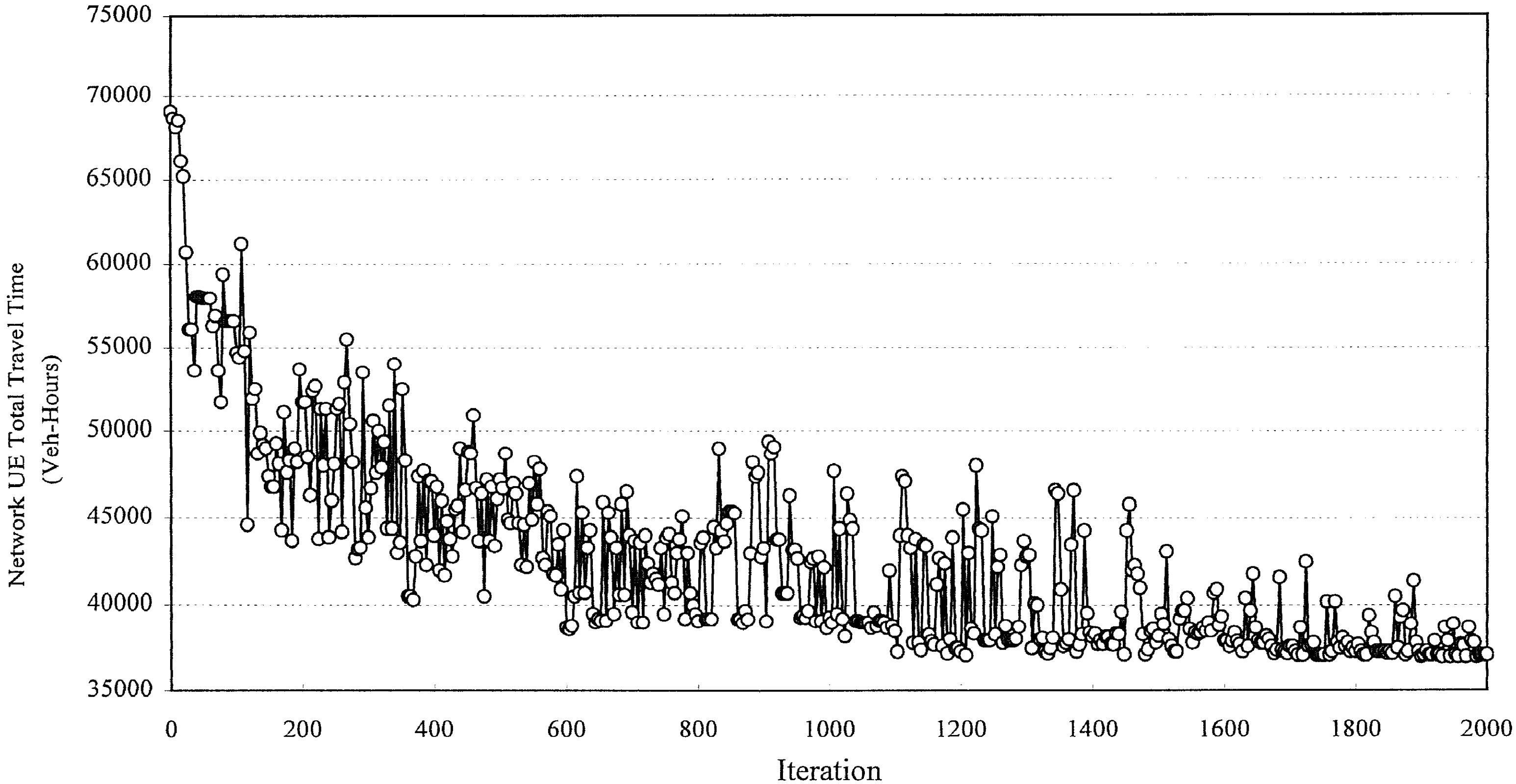


Figure 6.4.3 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, SA in Solving the TCNDP, Trial Solution State)

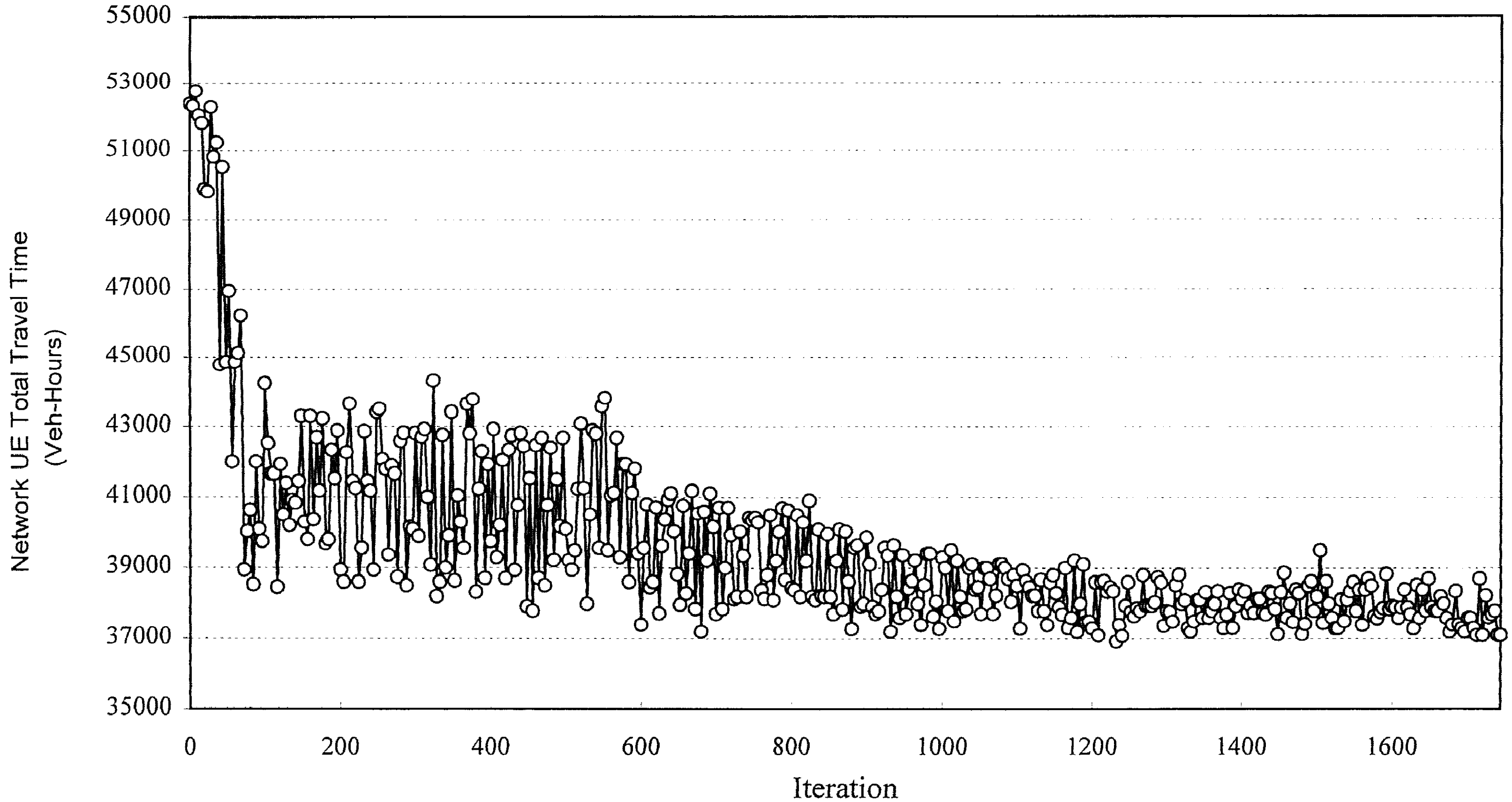


Figure 6.4.4 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, SA-TABU in Solving the TCNDP, Trial Solution State)

6.5 Sensitivity Analysis of the Key Components of the SA-TABU Search Strategy

The primary goal of this section's numerical experiments is to conduct sensitivity analysis of some key components of the SA-TABU search strategy, namely, Markov chain length, control parameter, tabu list length and the HEF.

The SA-TABU strategy is composed of three major sub-processes - simulated annealing, tabu search and the HEF. For the simulated annealing search strategy, the control parameter or "temperature" decreasing rate, and the Markov chain length are the two most important components that determine the efficiency and effectiveness of the search. A small temperature decreasing rate and a longer Markov chain length would yield a more smooth "annealing" that would generate the desired near-optimal solution, but it requires longer processing time. Thus, it is necessary to study the impact of these two parameters on the search procedure.

The search strategy uses a tabu search procedure to prevent the occurrence of cycling and to avoid local optimal. The tabu search procedure is implemented by not allowing the most recently updated element(s) to have its value changed for a period of time (number of iterations) that is termed as tabu list length. The tabu list length is the primary factor which features the tabu search characteristics of the algorithm.

In an heuristics search strategy, the most important factor is the heuristics evaluation function (HEF). In SA-TABU, a linear combination of the current link volume to capacity ratio (V/C ratio), the historical link performance and a random variable is used as the HEF. The current link volume capacity ratio, is one of the best parameter in characterizing the performance a link of the network. The link's LCOF variable provides the link's historical contribution to the objective function. The random variable transforms the HEF from a deterministic to a stochastic function. The form of this HEF and particularly the coefficients (weights) for each of the three components of the function have an impact on the performance of the algorithm.

Therefore, the sensitivity analysis focused on the above three components. The analysis was conducted on the five test networks, which were subjected to three different budget levels 10% , 20% and 30%.

6.5.1 Sensitivity Study on Markov Chain Length and Control Parameter

In the standard version of the SA-TABU search strategy, the control parameter decreasing rate is 15% and the Markov Chain length increasing rate is 20%. Therefore, 5% and 25% control parameter decreasing rates were used for comparison, while the Markov chain length increasing rate were set at 30% and 10%, respectively.

The Version 1- SA-TABU search strategy uses a 30% Markov chain length increasing rate and a 5% "temperature" dropping rate as a modification to the standard version, while Version 2 applied a 10% Markov chain length increasing rate and a 25% "temperature" dropping rate. In comparison to the standard SA-TABU version, a slower approach trend is expected to have a major impact on the Version 1 SA-TABU search strategy. Extending the Markov chain length under the same "temperature" allows the algorithm to accept more trial solution states and decreasing the "temperature" dropping rate also makes trial solution state acceptance criteria less restrictive. Consequently, the algorithm would explore a wider solution space and the search then resembles a more global search rather than a narrow local search. However, the slow progress creates a significant disadvantage on the efficiency of the algorithm, especially when the network size increases and the maximum iteration criterion is imposed. In contrast, the Version 2 - SA-TABU search strategy is much more restrictive on the acceptance of the trial solution state and approaches fast to a good solution, however, with an increasing risk of loosing good solutions.

The numerical results are summarized in Table 6.5.1. Figures 6.5.1-2 and Figure 6.4.4 present the trial solution state values during each iteration for each of the three versions of the SA-TABU as applied on the Network 3 30% budget level problem.

1) In the “best solution” category, Version 1 performs best for the small networks such as Network 1 and Network 2, where the feasible solution space is relatively small and the slow approach helps the search to explore as many feasible solutions as possible. As the network size increases, in most of cases, Version 2 produces the best “best solution”, and Version 1 is usually unable to reach convergence which implies it needs much more iterations in order to obtain a better “best solution”.

2) Version 2 requires the least number of iterations to converge since the approach focuses more on a smaller solution space of higher quality. Its ability to concentrate more on solution states of higher quality, makes the procedure more attractive for solving large scale networks, converging to better solution states at a few iterations.

Table 6.5.1 Summary of the Markov Chain Length and Control Parameter Sensitivity Analysis

NETWORK	BUDGET	BEST SOLUTION (Vehicle Hours) COMBINED ALGORITM VERSION1	VERSION0	VERSION3	ITERATIONS COMBINED ALGORITM VERSION1	VERSION0	VERSION3	MEAN OF TRIAL SOLUTION STATE (Vehicle Hours) COMBINED ALGORITM VERSION1	VERSION0	VERSION3	STANDARD DEVIATION OF TRIAL SOLUTION STATE (Vehicle Hours) COMBINED ALGORITM VERSION1	VERSION0	VERSION3
1	10%	6,599	6,605	6,875	1,218	788	541	7,122	7,117	7,047	133	129	94
	20%	6,488	6,488	6,488	2,000	1,002	866	6,803	6,765	6,759	192	189	165
	30%	6,232	6,232	6,232	1,535	310	288	6,311	6,269	6,251	121	101	53
2	10%	2,083	2,171	2,171	2,000	812	754	2,263	2,249	2,240	57	48	33
	20%	1,720	1,720	1,728	1,834	1,208	1,023	2,097	2,025	1,992	190	186	84
	30%	1,693	1,693	1,702	2,000	1,630	1,197	1,900	1,805	1,767	199	184	39
3	10%	38,434	38,584	38,699	2,000	1,536	1,567	46,121	44,878	44,993	1,872	1,662	1,684
	20%	37,963	37,024	37,072	2,000	1,028	1,025	43,616	41,460	41,747	1,473	1,287	1,269
	30%	37,012	36,920	36,928	2,000	1,756	1,665	42,742	41,509	39,334	2,775	2,568	1,770
4	10%	9,997	10,520	10,212	2,000	983	1,134	11,908	11,471	11,423	883	859	707
	20%	9,824	9,756	9,887	2,000	2,000	1,819	11,732	11,296	11,004	700	693	524
	30%	9,471	9,343	9,566	2,000	1,752	2,000	11,430	10,969	10,535	792	774	579
5	10%	23,914	23,512	24,700	2,000	2,000	1,784	44,297	39,762	38,298	19,922	18,878	16,135
	20%	21,512	20,005	21,180	2,000	2,000	2,000	41,006	39,623	37,000	16,718	15,333	14,792
	30%	18,269	18,269	18,269	2,000	2,000	2,000	33,123	28,936	24,394	11,025	9,126	8,642

Note:

Version 1: Same as Standard Version, except 5% - control parameter decreasing rate, 30%- Markov Length increasing rate.

Version 0: Standard Algorithm . See Section 6.4.1 for more details.

Version 3: Same as Standard Version, except 25% - control parameter decreasing rate, 10%- Markov Length increasing rate.

Table 6.5.2 Summary of Tabu List Length Sensitivity Analysis

NETWORK	BUDGET	BEST SOLUTION (Vehicle Hours) COMBINED ALGORITM			ITERATIONS COMBINED ALGORITM			MEAN OF TRIAL SOLUTION STATE (Vehicle Hours) COMBINED ALGORITM			STANDARD DEVIATION OF TRIAL SOLUTION STATE (Vehicle Hours) COMBINED ALGORITM		
		VERSION3	VERSION0	VERSION4	VERSION3	VERSION0	VERSION4	VERSION3	VERSION0	VERSION4	VERSION3	VERSION0	VERSION4
1	10%	6,605	6,605	6,814	939	788	684	7,215	7,117	7,008	102	129	131
	20%	6,488	6,488	6,488	1,148	1,002	1,139	6,774	6,765	6,843	174	189	148
	30%	6,232	6,232	6,279	556	310	407	6,333	6,269	6,508	93	101	125
2	10%	2,171	2,171	2,203	1,906	812	628	2,410	2,249	2,245	154	48	167
	20%	1,720	1,720	1,720	2,000	1,208	2,000	2,044	2,025	2,103	112	186	196
	30%	1,704	1,693	1,693	2,000	1,630	1,809	1,930	1,805	1,894	130	184	198
3	10%	39,781	38,584	38,584	1,688	1,536	1,076	45,031	44,878	43,004	1,537	1,662	2,131
	20%	37,048	37,024	37,024	1,290	1,028	1,544	40,539	41,460	41,824	1,624	1,287	1,857
	30%	37,003	36,920	36,920	1,980	1,756	2,000	39,874	41,509	39,999	1,972	2,568	2,815
4	10%	10,428	10,520	9,997	864	983	1,353	11,109	11,471	11,882	477	859	653
	20%	9,847	9,756	9,743	1,749	2,000	1,904	10,903	11,296	11,518	732	693	719
	30%	9,343	9,343	9,350	2,000	1,752	2,000	10,485	10,969	11,304	706	774	725
5	10%	23,690	23,512	23,523	2,000	2,000	2,000	39,808	39,762	39,415	14,246	18,878	149,765
	20%	21,188	20,005	20,003	2,000	2,000	2,000	39,621	39,623	38,994	15,713	15,333	15,763
	30%	19,715	18,269	18,269	2,000	2,000	2,000	26,737	28,936	27,469	7,049	9,126	10,027

Note:

Algorithm Version 3: Same as Standard Version, except 1/3 of Total Number of Links as Tabu Length.
Algorithm Version 0: Standard Algorithm . See Section 6.4.1 for more details..
Algorithm Version 4: Same as Standard Version, except 2/3 of Total Number of Links as Tabu Length.

Table 6.5.3 Summary of Heuristics Function Sensitivity Analysis

NETWORK	BUDGET	BEST SOLUTION (Vehicle Hours) COMBINED ALGORITM			ITERATIONS COMBINED ALGORITM			MEAN OF TRIAL SOLUTION STATE (Vehicle Hours) COMBINED ALGORITM			STANDARD DEVIATION OF TRIAL SOLUTION STATE (Vehicle Hours) COMBINED ALGORITM		
		VERSION5	VERSION0	VERSION6	VERSION5	VERSION0	VERSION6	VERSION5	VERSION0	VERSION6	VERSION5	VERSION0	VERSION6
1	10%	6,605	6,605	6,599	998	788	502	7,392	7,117	7,004	204	129	112
	20%	6,488	6,488	6,488	1,037	1,002	814	7,238	6,765	6,753	217	189	94
	30%	6,232	6,232	6,232	1,128	310	303	6,430	6,269	6,260	193	101	21
2	10%	2,232	2,171	2,083	2,000	812	677	2,251	2,249	2,241	92	48	47
	20%	1,720	1,720	1,785	1,818	1,208	1,028	2,084	2,025	1,923	188	186	178
	30%	1,704	1,693	1,693	2,000	1,630	930	1,856	1,805	1,793	101	184	165
3	10%	39,041	38,584	38,572	2,000	1,536	1,405	48,723	44,878	42,230	1,873	1,662	1,458
	20%	37,018	37,024	37,024	2,000	1,028	808	43,554	41,460	41,249	1,924	1,287	1,279
	30%	36,979	36,920	36,920	1,855	1,756	1,223	41,839	41,509	39,541	2,960	2,568	1,364
4	10%	10,391	10,520	10,579	2,000	983	767	11,672	11,471	11,302	1,001	859	753
	20%	9,788	9,756	9,740	2,000	2,000	2,000	11,587	11,296	11,290	832	693	565
	30%	9,462	9,343	9,343	2,000	1,752	2,000	11,401	10,969	10,904	887	774	712
5	10%	23,585	23,512	23,504	2,000	2,000	2,000	45,666	39,762	37,988	20,903	18,878	14,708
	20%	21,924	20,005	19,993	2,000	2,000	2,000	42,180	39,623	36,329	18,761	15,333	13,956
	30%	19,643	18,269	18,269	2,000	2,000	2,000	35,753	28,936	25,263	14,250	9,126	8,086

Note:

Version 5: Same as Standard Version, except 0.5 for the weight of V/C ratio is the heurisitcs function.

Version 0: Standard Algorithm . See Section 6.4.1 for more details.

Version 6: Same as Standard Version, except 1.5 for the weight of V/C ratio is the heurisitcs function.

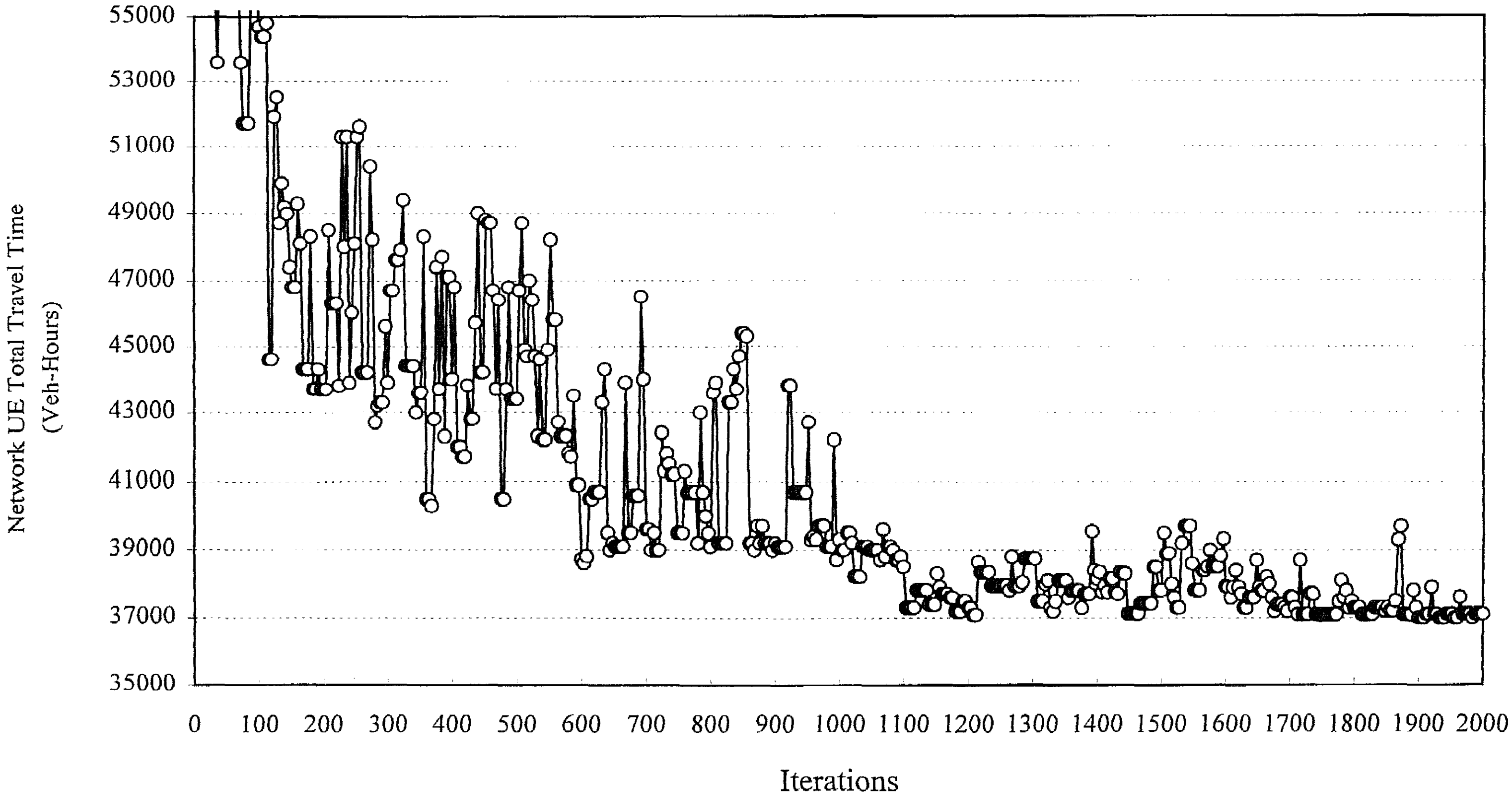


Figure 6.5.1 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, SA-TABU(Version 1) in Solving the TCNDP, Trial Solution State)

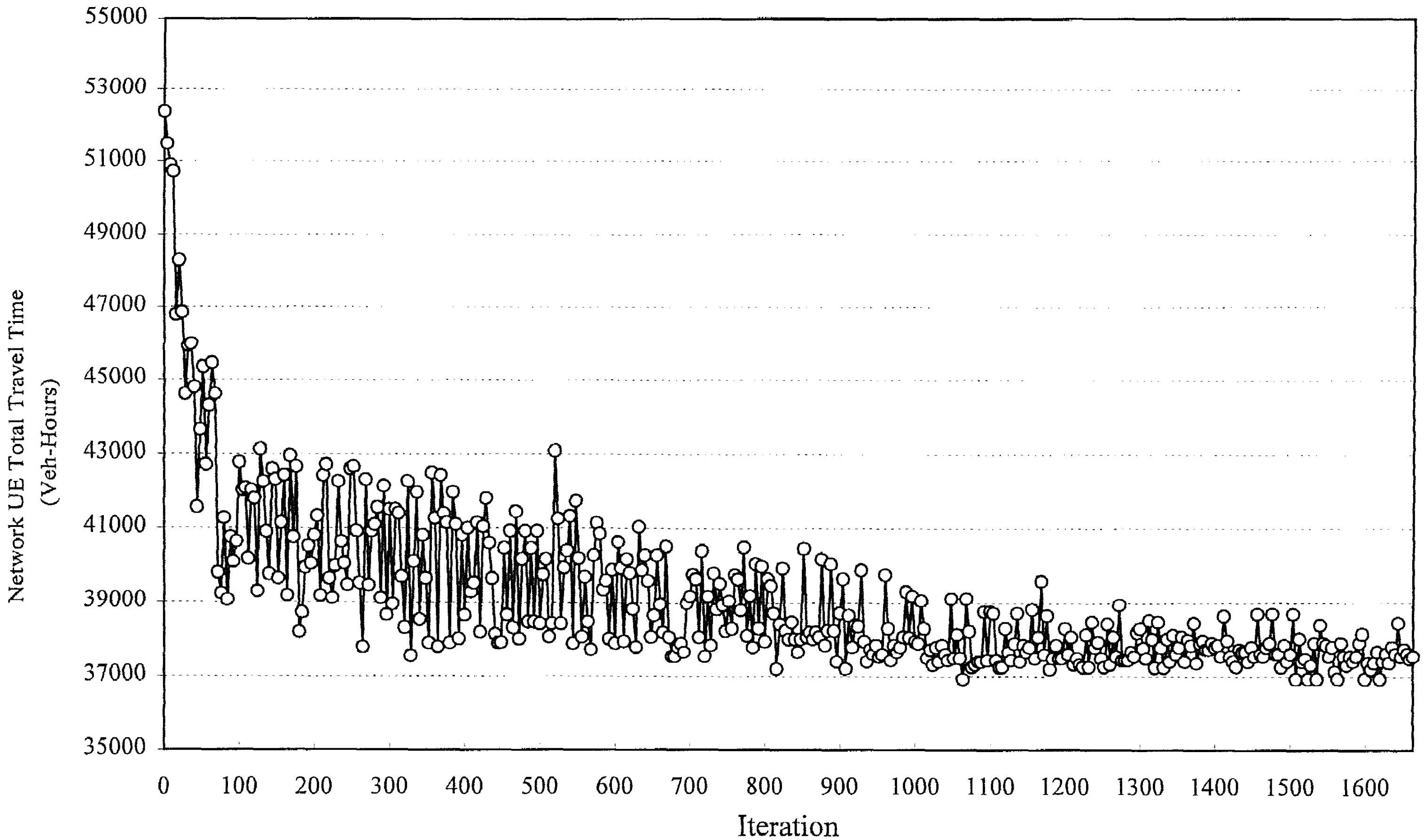


Figure 6.5.2 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, SA-TABU(Version 2) in Solving the TCNDP, Trial Solution State)

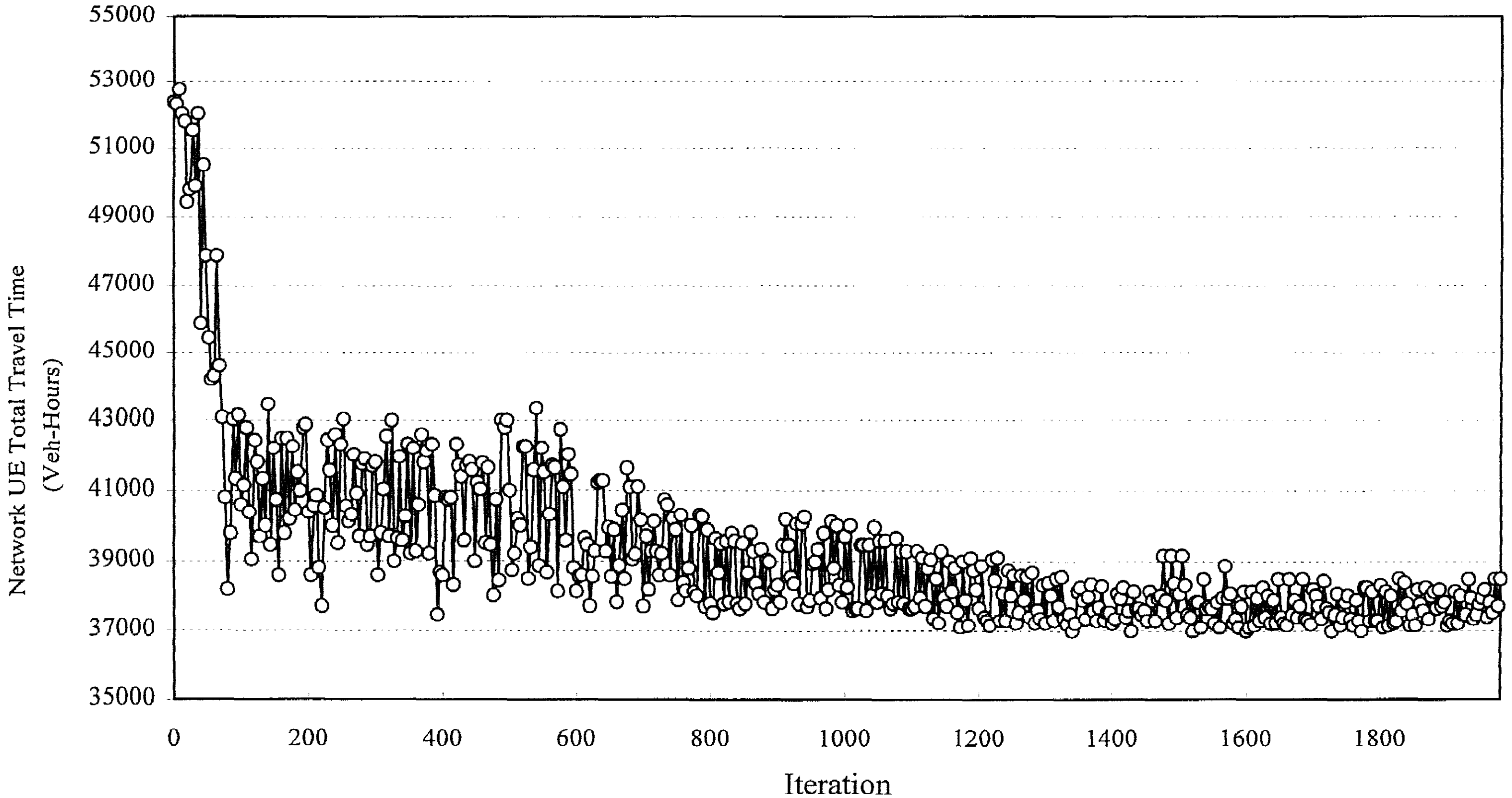


Figure 6.5.3 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, SA-TABU(Version 4) in Solving the TCNDP, Trial Solution State)

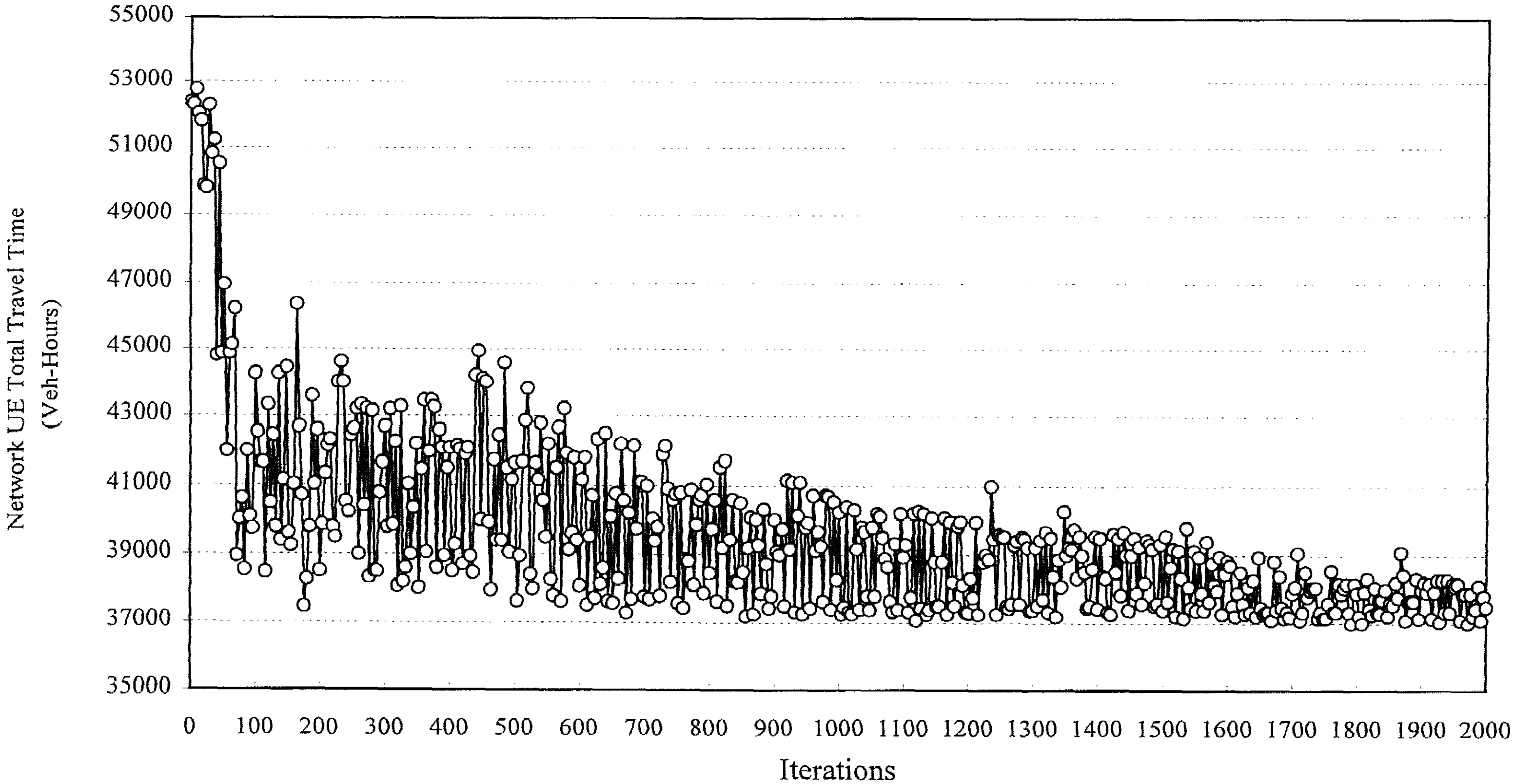


Figure 6.5.4 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, SA-TABU(Version 4) in Solving the TCNDP, Trial Solution State)

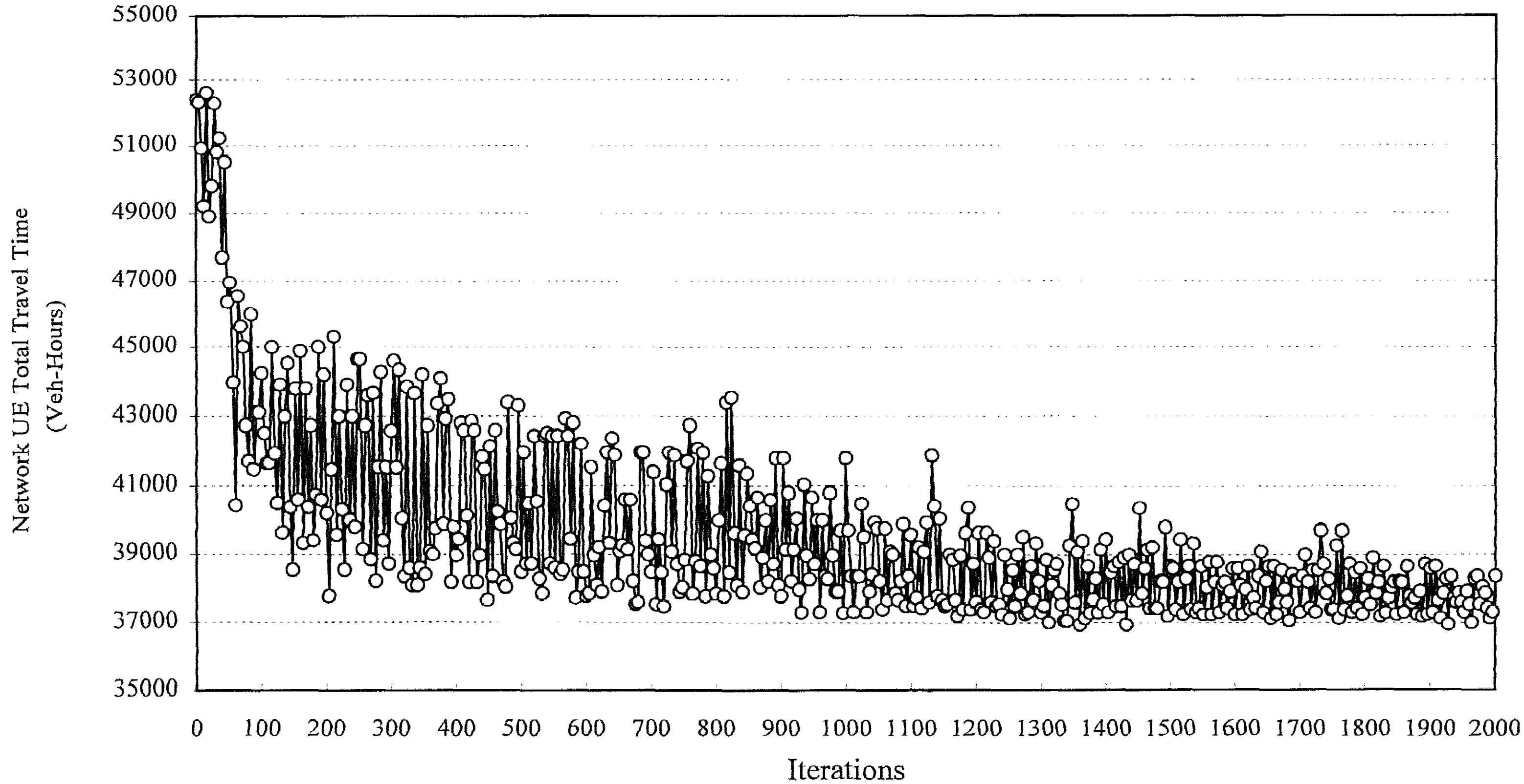


Figure 6.5.5 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, SA-TABU(Version 5) in Solving the TCNDP, Trial Solution State)

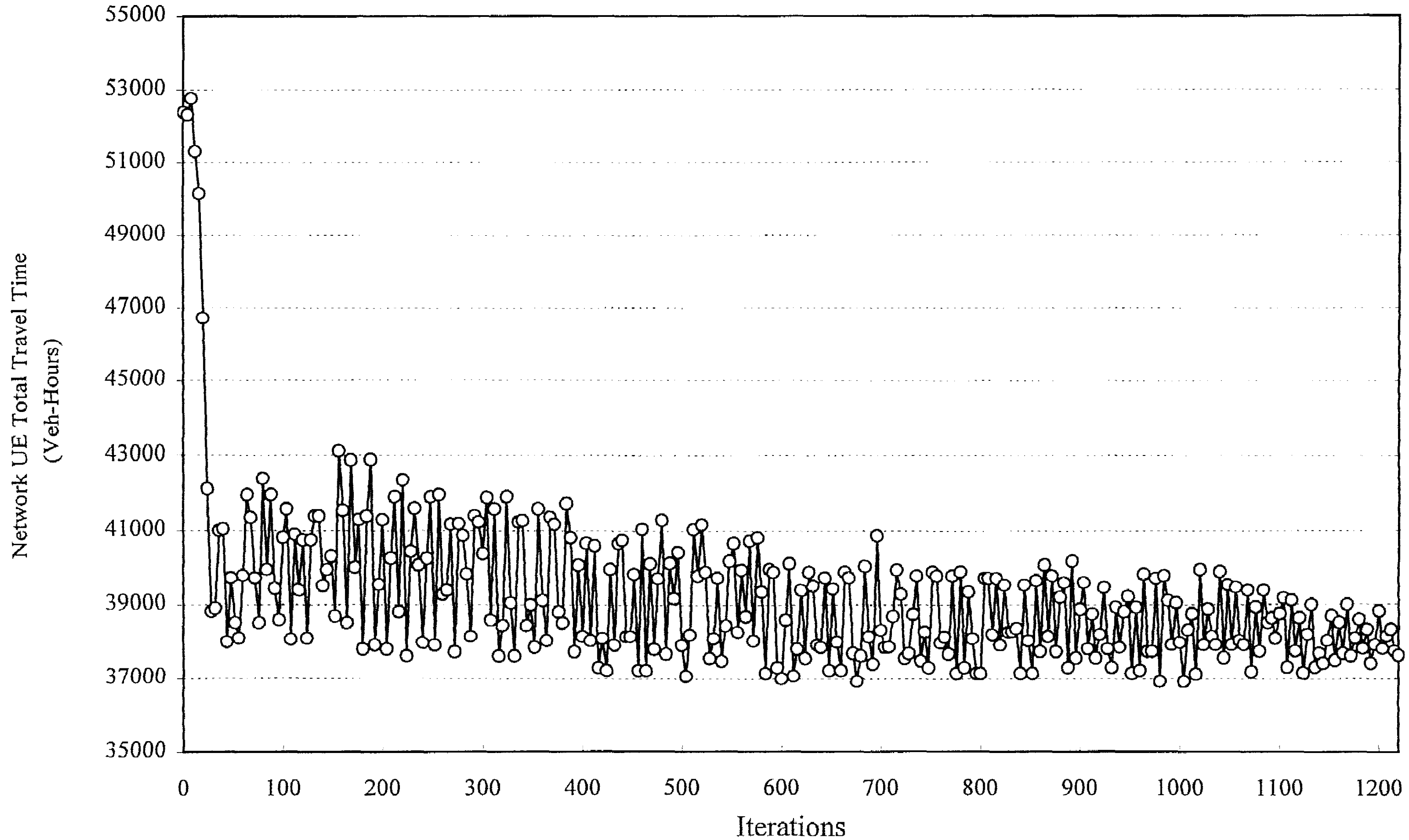


Figure 6.5.6 Network UE Total Travel Time (Veh-hours) VS. Iteration Number;(Network 3, 30% Budget Level, SA-TABU(Version 6) in Solving the TCNDP, Trial Solution State)

3) The combination of the trial solution state's mean and its standard deviation best describe the solution space the search strategy explores. It can be observed that Version 1 explores larger search space than Version 2 and the standard version. In the small size networks, Version 1 is able to cover most of the feasible solution space, while for larger networks, the feasible solution space coverage shrinks dramatically and the "good" solution space neighborhood is not explored enough. On the other hand, the Version 2 SA-TABU exhibits opposite characteristics to the Version 1.

It may be concluded that Version 1 is good for small network size problem and Version 2 is more suitable for large network size problems. Thus, the Markov chain length can be used in conjunction with the "temperature" dropping rate, as the control parameters in designing different versions of the algorithm according to the specific size of network problem. In this regard, the SA-TABU search strategy is a very flexible methodology which can control the solution quality and computational speed to fit different variety of problems.

6.5.2 Sensitivity Analysis on Tabu List Length

In the simple tabu search, the tabu list length has a major effect on the algorithm. In Glover(1990) a tabu list length 7 is suggested. However, the proper tabu list length also depends upon the problem size. The standard version algorithm adopts half of the total number of links as the tabu list length. The sensitivity analysis utilizes 33% and 67% of the total number of the links as the tabu list lengths for comparison.

The tabu list length of the Version 3 -SA-TABU is 33% of the total number of the links, and in Version 4 -SA-TABU, the tabu list length is 67% of the total number of the links. When the tabu list length is shorter, the algorithm would allow more appearances of the same solution states that are "good" solution states. However, the generated trial solution state set to be explored is smaller. In this aspect, the SA-TABU with shorter tabu list length has similar characteristics with the shorter Markov Chain length or higher "temperature" rate such as Version 1-SA-TABU. However, the search procedures based on tabu list length variations are not equivalent to the algorithms on Markov chain length or "temperature" dropping rate variations, since they are more related to the size of the explored trial solution set, while the later ones refer to the acceptance of the trial solution states. A summary table of the experimental results is presented in Table 6.5.2 and in Figures 6.5.3-4, and Figure 6.4.4 of the Network 3 30% budget level problem. The results are summarized below:

1) In Network 1 and Network 2 problems, where the problem size is relative small, the Version 3 produces the best "best solution" and the Version 4 the worst "best solution". This may be caused by the longer tabu list length used by the Version 3 -SA-TABU, as the "good" set of solution states is not explored enough. This is also evidenced by the required number of iterations. The Version 4 -SA-TABU search requires the fewest number of iterations, especially for Network 1, due to the fact that a larger number of "good" solution states are held on the tabu list and the acceptance of the trial solution state becomes harder, resulting in the early termination of the process. This also causes the mean of the trial solution state for the Version 4-SA-

TABU to be relative higher. Furthermore, the strict acceptance of the trial solution state leads to a higher standard deviation of the trial solution state means.

2) For relative large problem sizes such as Network 3, 4 and 5, the Version 4 -SA-TABU uses less iterations to obtain a better "best solution". Network 3, 4 and 5 problems also have less congested network flow, and the sets of "good solution" are larger. The longer the tabu list length is, it helps the search to explore more of the "good solution" set by diversifying the search. Based on the trial solution state mean and standard deviation of the trial solution state, the Version 4 search strategy demonstrated its ability of stretching out the search space under the "good" solution state neighborhood. The reason for Version 2's lackluster performance in the large scale problems is that the short tabu list length, results in the frequent occurrence of cycling, visiting the same solution states, while losing valuable processing time to explore the new feasible solution space.

In conclusion, the tabu list length is one of the vital elements that greatly affects the quality of the search procedure. It should be designed according to the characteristics of the specific problems. The tabu list length should be longer if the set of "good solutions" is relatively large. The size of the "good solution" set depends upon the size of the network and the number of similar links having very similar HEF values in the network.

6.5.3 Sensitivity Analysis on Heuristic Evaluation Functions (HEF)

There are three variables in the HEF, the V/C ratio, the basic heuristic variable and random number. The weight for V/C ratio variable is 1, LCOF's value was set to 0.5 and for the random "error" was set to 1. Early experiments showed that the performance of the procedure does not have a clear relation with the variation of the weight of the basic heuristics value variable. In these experiments, the weight factor ratio for the V/C ratio versus the random variable is explored.

The weight of the V/C ratio variable for the Version 5 - SA-TAWBU procedure is 0.5 and 1.5 for the Version 6 procedure. In essence, the Version 6 procedure relies more on the heuristic information to direct the search. The use of the heuristic information guides the search towards a "good solution" state neighborhood that makes the procedure more efficient and effective. However, the excessive use of the heuristic information may force the search to avoid some very "good" solution states or the global optimum.

The previous experiments provided some insights into the elements that play an important role to the performance of the search strategy, such as the Markov chain length, and the tabu list length, should be designed to reflect the specific problem characteristics. The variation on the Markov chain length and the "temperature" dropping rate dictate the move size, and the tabu list forces the selection of non-prohibited moves, while the V/C ratio utilizes the network flow characteristics.

Table 6.5.3 presents a summary of the experimental results for Versions 5, 6 and the standard version procedures, while Figures 6.5.5-6 depict the trial solution state values versus the number of iterations for the Version 5 and 6 procedures, respectively. The following observations are made:

1) Version 6 procedure requires the least number of iterations, the smallest mean and standard deviation of the trial solution states in Network 1 and Network 2, respectively. The "best solution" does not follow any trend by the problem size. The use of the V/C ratio does not provide any advantage to the search procedure, based on the "best solution" observed. Using a higher weight of the V/C ratio information, Version 6 focuses more on some "elite" links resulting in less iterations and a smaller trial solution states space.

2) Networks 3, 4 and 5 are larger than Networks 1and 2. Version 6 has an advantage over the other two versions of the searches, for the larger networks, by reaching the best "best solution" in less iterations and exploring a smaller trial solution state space. However, it is not necessary true that the larger the problem size is, the more the procedure relies on the V/C ratio information. It can only be concluded that the search procedure utilizing higher V/C ratio information , forces the search towards "good solution" trial solution state sets, which might be a local optimum, thereby increasing the risk of missing the global optimum.

Identifying the proper weights for the HEF for a specific problem is much more difficult than the Markov chain length , "temperature" dropping rate and the tabu list length which can be derived from the size of the problem. However, the higher V/C ratio weight tends to be more suitable for large networks if finding the global optimum is not strictly required and the computing time is constrained.

The series of experiments conducted provided a general understanding of the SA-TABU search strategy in solving the TCNDP. The procedures presented are highly stochastic due to the two times that random variables are involved (One appears in the acceptance criteria and the other is involved in the HEF), which do not provide any specific conclusions. However, some general conclusions are derived which facilitated the decisions on the use of certain weights on the variables of the HEF.

It can be concluded that SA-TABU search strategy is not only an efficient and feasible methodology in solving large scale TCNDP, but it also a very flexible and robust algorithm. It can be designed to fit a specific problem by modifying the HEF, the iteration processing manner and trial solution state size to reflect different problem structures, characteristics, size and complexity.

CHAPTER 7

TWO CLASSES OF USERS NETWORK DESIGN PROBLEM STUDY

This chapter presents the results of numerical experiments on network flow characteristics and their implications in network design. A study of the characteristics of the TCNDP is also presented.

7.1 Study of the Characteristics of the TCNDP

The network design variables considered in this study involve the addition of an extra lane to the existing network links which can assume one of three traffic operation options: i) extra lane is allowed for both passenger car and truck; ii) extra lane is exclusively for the trucks; iii) extra lane is exclusively for the passenger cars. In this section, a simplified example is utilized to better illustrate how the two classes of users respond to the addition of an extra lane, subject to the specific traffic operational options specified above.

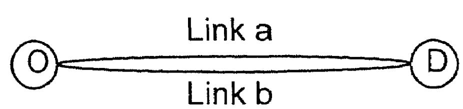


Figure 7.1.1 Example Network

Figure 7.1.1 shows a simple one link network. It is assumed that the passenger car link cost function is: $t_a = 2(1+1.03(\frac{x_a + 2x_b}{1600})^{5.52})$, and the truck link Link b cost function as: $t_b = 3(1 + 0.62(\frac{0.5x_a + 4x_b}{1600})^{5.14})$. The passenger car demand is 1200 and the truck demand is 200.

Under the scenario with an exclusive passenger car lane and truck lane, for the roadway is utilized by the same class of user, it is assumed that the travel speed in the new lane would increase by 10%. The streamlined diagonalization algorithm, is used to assign the traffic flow of the two classes of users over the original network and the expanded network under the three different options. The results are shown in Table 7.1.1.

From Table 7.1.1, the following observations are made:

i) The addition of a new lane improves the total travel time. The three different types of lane addition produced better total UE travel times compared to the original network. The class's link travel time in every scenario has decreased by the reduction of its own congestion or its counterpart's traffic congestion.

ii) The link travel time for each class is identical between the existing lane and the new lane. This confirms that this is an equilibrium solution.

iii) The exclusive lane for either class not only significantly improves its own user's travel time, but also it decreases its counterpart's travel time by a certain percentage. The counterpart's percentage decrease is less than the corresponding decrease observed under the shared lane option expansion. The option for expansion of the capacities for both classes (shared lane) equally improves the passenger car and truck's travel time.

Table 7.1.1 Computational Results of the Example in Figure 7.1.1

	ORIGINAL NETWORK	NEW LANE FOR BOTH (% Reduction)	EXCLUSIVE NEW CAR LANE (% Reduction)	EXCLUSIVE NEW TRUCK LANE (% Reduction)
TOTAL TRAVEL TIME(veh-min)	5659	3287 (42%)	3246 (43%)	3810 (33%)
Existing Car Lane				
Travel Time(minutes)	3.94	2.22 (44%)	2.16 (45%)	2.67 (32%)
Flow(cars)	1200	800	606	1200
Existing Truck Lane				
Travel Time(minutes)	4.06	3.12 (23%)	3.27 (19%)	3.06 (25%)
Flow(trucks)	200	133	200	52
New Car Lane				
Travel Time(minutes)		2.22	2.16	
Flow(cars)		400	594	
New Truck Lane				
Travel Time(minutes)		3.12		3.06
Flow(trucks)		67		148

iv) The exclusive car lane option produces the optimal total travel time in comparison to the two options. The passenger car traffic demand accounts for 87.5% of the total flow. The addition of the passenger car lane produces the largest improvement for the majority of users, the passenger cars, so the total travel time saved is maximized.

v) These results may also have policy implications. For example, if it is desired to benefit either the trucks or passenger cars, then an exclusive new truck or car lane should be sought. If the objective is to minimize the total travel time then a new exclusive car lane should be constructed. If it is desired to reduce the travel time for both classes in an equitable manner, then a shared lane would be appropriate.

It is noted however, that the above results are applicable to the specific example and the specific travel cost functions utilized. It cannot be concluded that they can be generalized to networks.

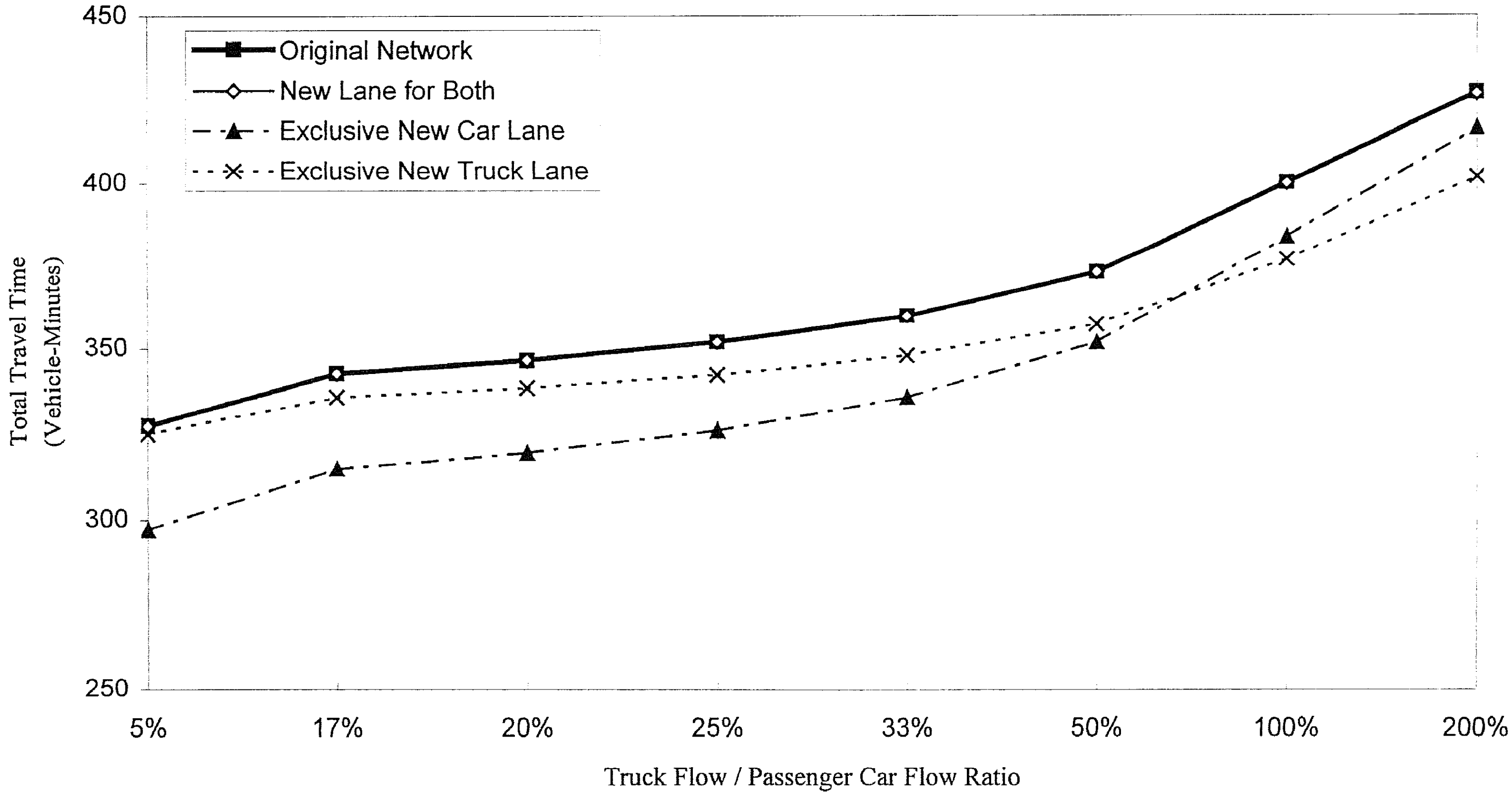


Figure 7.1.2 Total Network Travel Time vs. Truck /Car Flow Ratio; V/C ratio = 0.1

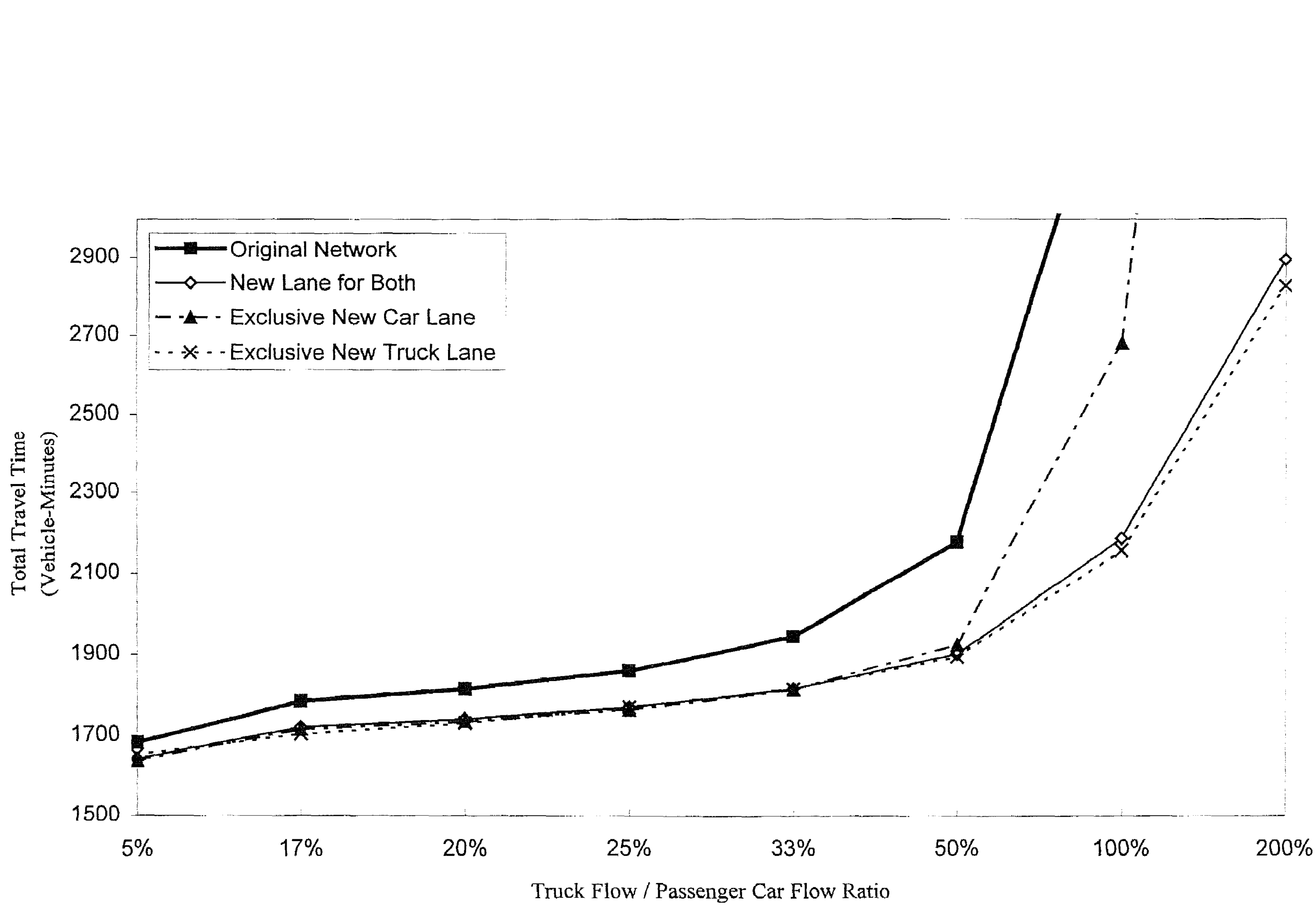


Figure 7.1.3 Total Network Travel Time vs. Truck /Car Flow Ratio; V/C ratio = 0.5

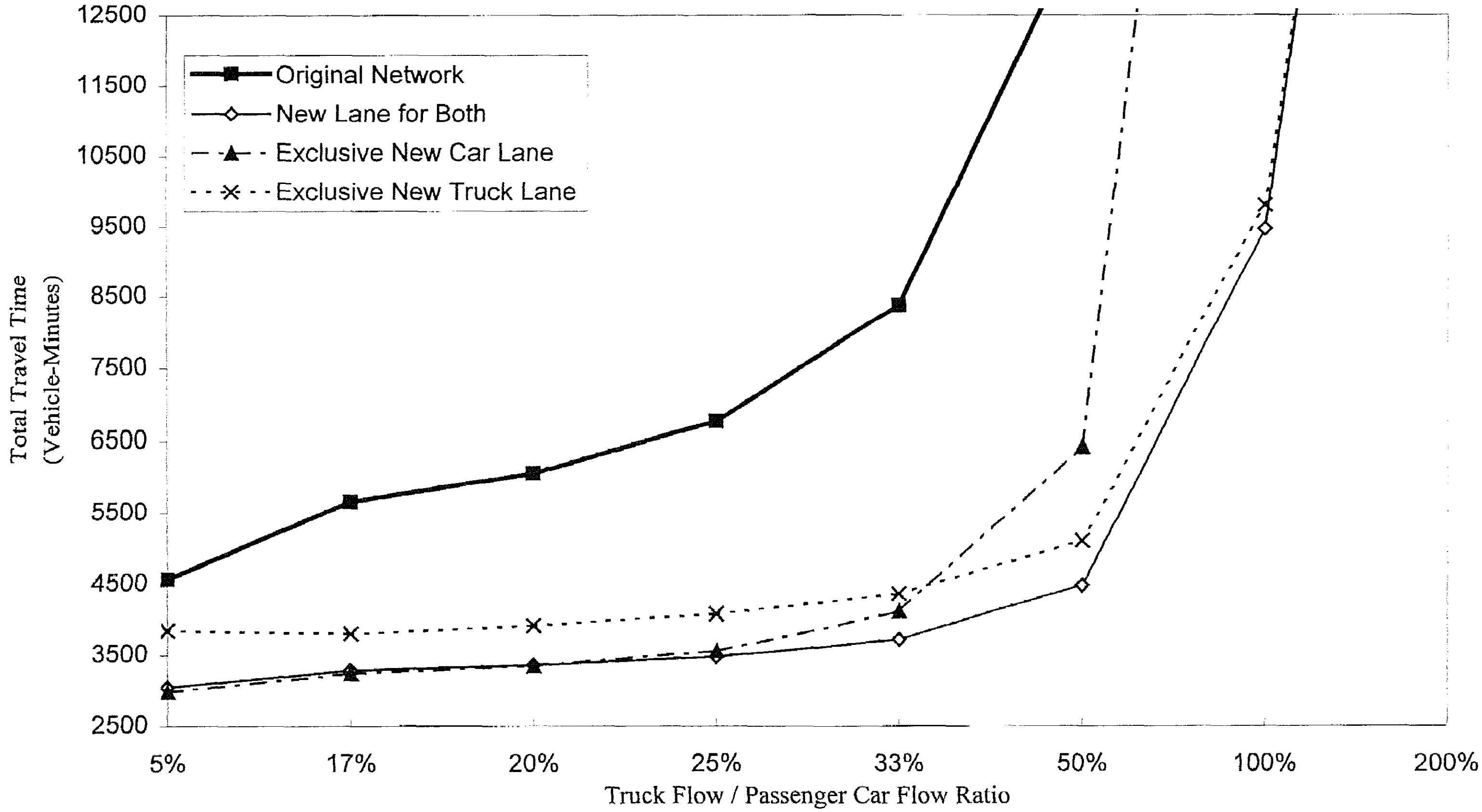


Figure 7.1.4 Total Network Travel Time vs. Truck /Car Flow Ratio; V/C ratio = 0.875

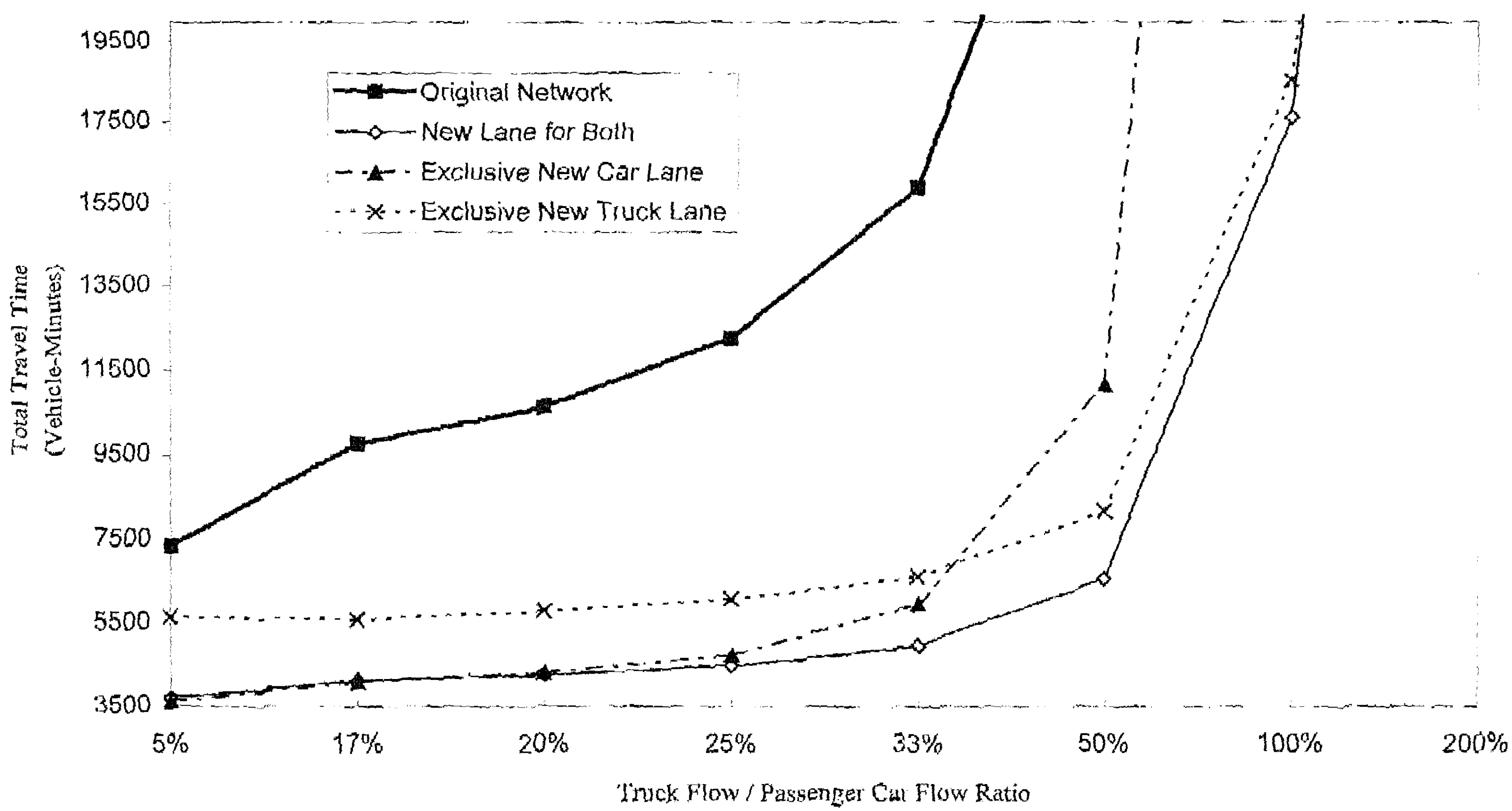


Figure 7.1.5 Total Network Travel Time vs. Truck /Car Flow Ratio; V/C ratio = 1.0

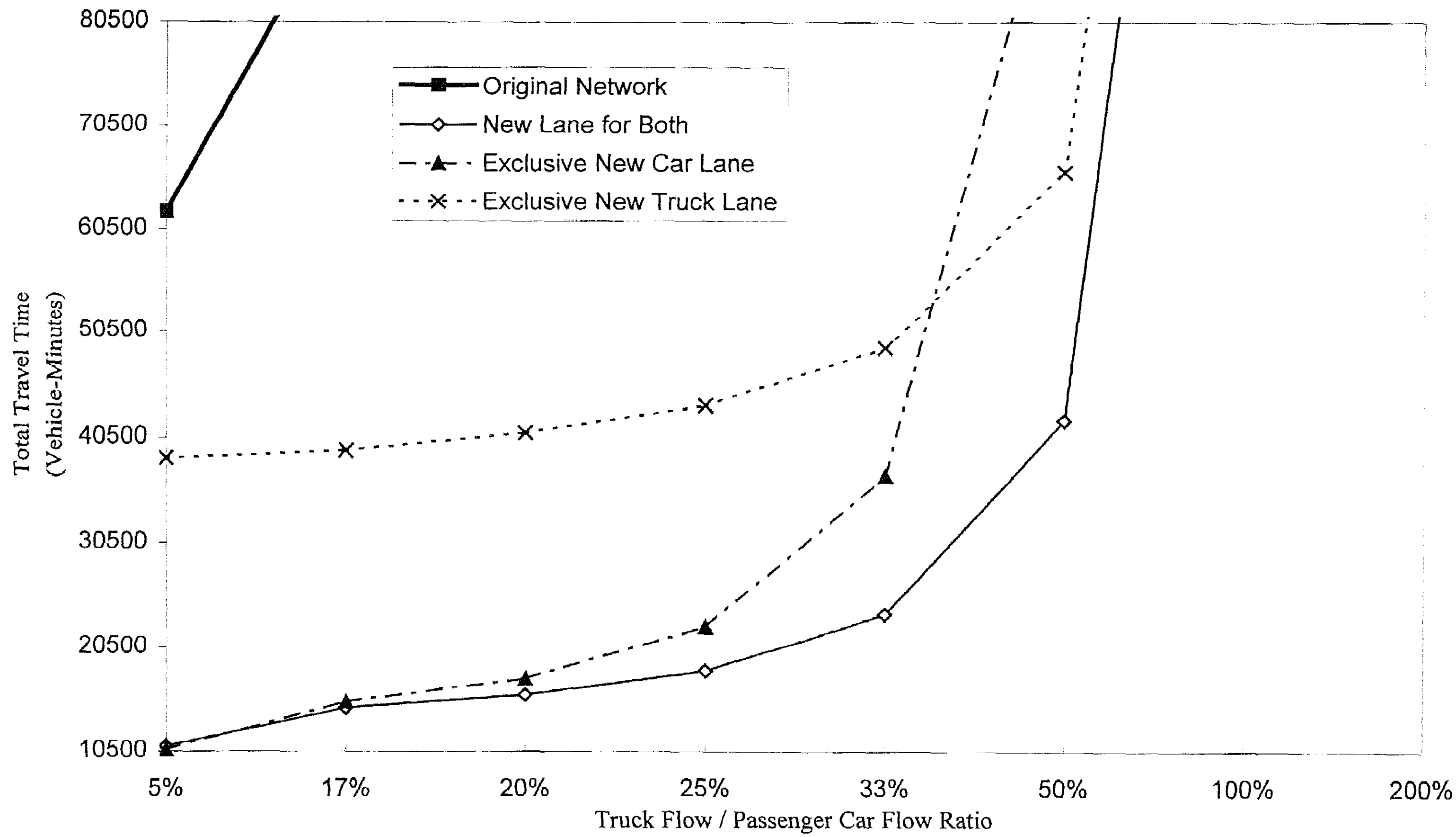


Figure 7.1.6 Total Network Travel Time vs. Truck /Car Flow Ratio; V/C ratio = 1.5

Using the same example, several experiments were conducted with different levels of congestion and truck to passenger car volume ratio (See Figures 7.1.2-7.1.6). In these figures, the horizontal axis shows the trucks to passenger cars volume ratio, varying from 5% to 200%, and the vertical axis is the total travel time, while the four different category lines represent the original network and three network expansion options, respectively. Figures 7.1.2-7.1.6, show the results under the 0.1, 0.5, 0.875, 1.0, 1.5 congestion levels (V/C ratio).

It is observed that the aforementioned conclusions obtained from the example were preserved in all test scenarios. The following observations are made:

i) Figure 7.1.2, with 0.1 V/C ratio, shows that the exclusive car lane performs best for truck/car flow ratios of less than 50%.

ii) In Figure 7.1.3, when the V/C ratio is 0.5, new lane for both, exclusive new car lane, and exclusive new truck lane produce the same improvement up to a truck/passenger car flow ratio of 50%. Then the shared lane and the truck lane perform best.

iii) Figure 7.1.4 demonstrates that shared lane and exclusive passenger car lane performs best up to a truck/car flow ratio of 25%, when the shared lane starts performing better. After about a 40% truck/car flow ratio, the exclusive car lane becomes the worst of the three options.

iv) A V/C ratio of 1, it produces similar results to those with a V/C ratio of 0,875, which is shown in Figure 7.1.5 and Figure 7.1.4.

v) Under the condition of V/C ratio equal to 1.5 of Figure 7.1.6, the shared lane produces the best results for almost all the truck/car flow ratios.

7.2 The Experimental Results of the Network Design Solutions

In this section, the network results such as network flow and network design solution are summarized.

Table 7.2.1 presents the network design solution and the network link flow volumes for Network 1 with 10%, 20% and 30% budget level with the best solutions of 6605, 6488 and 6232 vehicle-hours, respectively, of the network total UE travel time. The best total UE travel time for Network 2 (Table 7.2.2) were 2171, 1720 and 1693 vehicle-hours for the 10%, 20% and 30% budget levels, respectively. Similarly, Tables 7.2.3-7.2.5 demonstrate the best network solutions and the link flow volumes for Network 3, Network 4 and Network 5 with 10%, 20% and 30% budget levels, respectively.

Some key interesting observations can be derived from the results in Tables 7.2.1-7.2.5 as follows:

1) Almost every expanded link of the best solution is a shared lane (passenger cars and trucks) operations.
2) The links selected to be expanded at a lower budget level are not necessary selected at higher budget level. This is consistent to the findings for the SCNDP (Mouskos, 1991). This is expected as the flows redistributed themselves based on the new network configuration which is the essence of Braess's paradox.

Table 7.2.1 Network 1, TCNDP Best Solutions and Network Link Flows

NETWORK 1															
Network Configuration Link		**Capacity**	**Free Flow Speed**	**Initial Nework**			**10% Budget**			**20% Budget**			**30% Budget**		
From	*To*	*(Vehicles)*	*(Miles/Hour)*	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C
11	12	3000	50	3,395	314	1.55	3,802	371	1.76	3,423	320	1.57	3,209	258	1.41
11	15	1000	30	2,236	210	3.07	1,654	156	1.52	1,736	186	2.48	1,554	162	2.20
11	16	3750	45	4,836	484	1.81	4,257	425	1.59	5,300	521	1.97	5,617	547	2.08
12	11	3000	50	3,395	314	1.55	3,377	370	1.62	3,480	374	1.66	3,514	353	1.64
12	13	500	30	928	107	2.71	693	84	2.06	964	93	2.67	859	91	2.45
13	12	500	30	928	107	2.71	479	85	1.64	501	73	0.79	587	92	0.96
13	14	1500	35	2,489	273	2.39	1,996	262	2.03	1,752	186	1.25	1,710	177	1.21
13	16	1000	30	1,765	189	2.52	1,347	121	1.83	1,417	103	1.83	1,448	136	1.99
14	13	1500	35	2,489	273	2.39	2,070	257	2.07	2,267	275	2.25	1,783	228	1.35
14	15	3750	45	4,381	453	1.65	3,173	336	1.20	3,649	379	1.38	3,279	299	1.19
14	16	500	30	1,327	113	3.56	890	77	1.20	986	87	1.33	918	76	1.22
15	11	1000	30	2,236	210	3.07	1,450	142	2.02	1,620	161	2.26	1,502	140	1.38
15	14	3750	45	4,381	453	1.65	3,021	321	1.15	3,134	351	1.21	3,324	352	1.26
15	16	1000	30	2,039	210	2.88	1,303	116	1.77	1,509	151	1.41	1,378	118	1.85
16	11	3750	45	4,836	484	1.81	4,988	447	1.81	5,358	493	1.95	4,606	401	1.66
16	13	1000	30	1,765	189	2.52	1,650	125	2.15	1,838	162	2.48	1,651	153	2.26
16	14	500	30	1,327	113	3.56	1,003	81	1.33	1,071	98	1.46	948	88	1.30
16	15	1000	30	2,039	210	2.88	1,514	140	2.07	1,628	176	2.33	1,401	147	1.33

Note: The links in the shadow are the expanded links.

Table 7.2.2 Network 2, TCNDP Best Solutions and Network Link Flows

Network Configuration Link From	To	Capacity (Vehicles)	Free Flow Speed (Miles/Hour)	Initial Nework Car Flow (Vehicles)	Initial Nework Truck Flow (Vehicles)	Initial Nework V/C	10% Budget Car Flow (Vehicles)	10% Budget Truck Flow (Vehicles)	10% Budget V/C	20% Budget Car Flow (Vehicles)	20% Budget Truck Flow (Vehicles)	20% Budget V/C	30% Budget Car Flow (Vehicles)	30% Budget Truck Flow (Vehicles)	30% Budget V/C
9	10	500	30	583	29	1.40	592	28	1.41	402	49	0.60	467	32	0.59
9	18	4500	50	2,696	357	0.92	2,555	365	0.89	2,212	356	0.81	2,135	336	0.77
9	17	1500	35	1,221	34	0.90	1,353	27	0.97	1,384	15	0.96	1,430	21	1.01
10	9	500	30	586	30	1.41	576	30	1.39	399	49	0.60	421	48	0.61
10	11	1500	35	957	29	0.72	960	28	0.71	990	49	0.79	993	64	0.83
10	18	500	30	326	0	0.65	298	0	0.60	75	0	0.15	51	0	0.10
11	10	1500	35	911	30	0.69	874	30	0.66	980	49	0.78	986	48	0.78
11	12	4500	50	778	20	0.19	827	20	0.20	720	20	0.18	861	20	0.21
11	18	2000	40	1,311	410	1.48	1,299	410	1.47	1,300	391	1.43	1,153	392	1.36
12	11	4500	50	642	150	0.28	636	150	0.27	693	150	0.29	758	150	0.30
12	13	4500	50	2,245	270	0.74	2,096	270	0.71	1,822	270	0.64	2,210	270	0.73
12	18	2000	40	1,895	90	1.13	1,806	90	1.08	1,736	90	1.05	1,325	90	0.60
13	12	4500	50	2,343	240	0.73	2,108	240	0.68	2,149	240	0.69	2,251	240	0.71
13	14	1000	30	1,327	64	1.58	1,034	54	0.83	1,013	53	0.82	778	46	0.64
13	15	500	30	571	36	1.43	519	20	1.20	542	19	1.24	409	14	0.47
13	18	500	30	485	0	0.97	428	0	0.86	413	0	0.83	301	0	0.60
14	13	1000	30	1,409	11	1.45	1,019	12	0.71	992	15	0.70	793	10	0.55
14	15	500	30	76	0	0.15	143	0	0.14	263	0	0.53	12	0	0.02
14	16	1500	35	1,385	64	1.09	1,276	80	1.06	1,296	81	1.08	1,178	70	0.97
15	13	500	30	573	9	1.22	523	3	1.07	521	5	1.08	403	1	0.41
15	14	500	30	58	0	0.12	92	0	0.18	120	0	0.12	9	0	0.02
15	16	500	30	571	36	1.43	430	20	0.51	448	19	0.53	409	14	0.47
15	18	1500	35	105	0	0.07	107	0	0.07	132	0	0.09	17	0	0.01
16	14	1500	35	1,485	11	1.02	1,496	17	1.04	1,314	15	0.92	1,207	15	0.85
16	15	500	30	602	9	1.27	533	3	0.54	404	2	0.82	404	1	0.41
16	17	1500	35	1,585	69	1.24	1,284	41	0.72	1,283	16	0.90	1,265	37	0.94
16	18	2000	40	1,328	531	1.73	1,002	559	1.62	964	564	1.61	773	563	1.08
17	9	1500	35	1,109	69	0.92	1,370	41	1.02	1,372	36	1.01	1,190	37	0.89
17	16	1500	35	1,580	34	1.14	1,472	27	1.05	1,466	15	1.02	1,221	5	0.62
17	18	2000	40	476	0	0.24	599	0	0.30	549	0	0.27	75	0	0.04
18	9	4500	50	2,806	331	0.92	2,554	360	0.89	2,223	345	0.80	2,375	345	0.83
18	10	500	30	374	0	0.75	369	0	0.74	86	0	0.17	59	0	0.12
18	11	2000	40	1,401	391	1.48	1,404	392	1.49	1,317	371	1.40	1,249	356	1.34
18	12	2000	40	1,661	250	1.33	1,603	250	1.30	1,382	250	1.19	1,114	250	0.76
18	13	500	30	499	0	1.00	467	0	0.47	379	0	0.76	329	0	0.66
18	15	1500	35	58	0	0.04	236	0	0.16	194	0	0.13	9	0	0.01
18	16	2000	40	1,464	416	1.56	1,344	423	1.52	1,323	435	1.53	1,198	429	1.46
18	17	2000	40	358	0	0.18	119	0	0.06	82	0	0.04	378	0	0.19

Note: The links in the shadow are the expanded links.

Table 7.2.3 Network 3, TCNDP Best Solutions and Network Link Flows

NETWORK 3															
Network Configuration Link		**Capacity**	**Free Flow Speed**	**Initial Nework**			**10% Budget**			**20% Budget**			**30% Budget**		
From	*To*	*(Vehicles)*	*(Miles/Hour)*	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C
11	12	3000	50	1,514	103	0.64	1,629	127	0.71	1,662	120	0.71	1,615	106	0.68
11	15	3000	50	1,414	85	0.58	1,337	68	0.54	1,464	66	0.58	1,419	62	0.56
11	16	2500	45	1,605	157	0.89	1,674	145	0.90	1,593	146	0.87	1,625	62	0.75
12	11	3000	50	3,019	242	1.33	3,011	212	1.29	3,057	212	1.30	2,275	108	0.68
12	16	1500	30	1,506	151	1.41	1,484	158	1.41	1,480	156	1.40	1,152	181	0.82
12	13	3000	50	2,975	178	1.23	3,005	199	1.27	2,963	202	1.26	2,301	162	0.98
13	12	3000	50	1,946	97	0.78	1,696	73	0.66	1,688	80	0.67	1,723	94	0.70
13	14	3000	50	2,601	178	1.10	2,585	199	1.13	2,577	202	1.13	2,201	163	0.71
13	17	1500	30	374	0	0.25	420	0	0.28	385	0	0.26	123	0	0.08
14	13	3000	50	1,428	61	0.56	1,280	59	0.50	1,110	69	0.46	1,113	82	0.48
14	21	3000	50	1,690	197	0.83	1,621	225	0.84	1,604	229	0.84	1,819	233	0.92
14	17	2500	45	2,344	266	1.36	2,318	239	1.31	2,314	234	1.30	2,315	235	0.93
15	11	3000	50	1,514	103	0.64	1,629	127	0.71	1,662	120	0.71	1,615	106	0.68
15	18	3000	50	1,414	85	0.58	1,337	68	0.54	1,464	66	0.58	1,419	62	0.56
16	11	2500	45	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
16	12	1500	30	839	80	0.77	976	80	0.86	950	80	0.85	963	80	0.86
16	17	4500	30	1,793	141	0.52	1,664	141	0.50	1,525	108	0.43	1,477	110	0.43
16	18	2500	45	1,692	125	0.88	1,691	128	0.88	1,598	128	0.84	1,694	128	0.88
17	16	4500	30	2,894	334	0.94	2,363	292	0.78	2,570	289	0.83	2,309	305	0.78
17	13	1500	30	518	36	0.44	415	15	0.32	579	11	0.42	610	12	0.44
17	14	2500	45	3,576	220	1.78	3,609	231	1.81	3,454	230	1.75	3,516	226	1.26
17	20	2500	30	504	23	0.24	652	9	0.28	633	7	0.26	367	8	0.16
18	15	3000	50	1,514	103	0.64	1,629	127	0.71	1,662	120	0.71	1,615	106	0.68
18	16	2500	45	6,520	1,185	4.50	7,010	1,234	3.41	6,630	1,205	3.27	6,857	1,191	3.32
18	19	3000	50	269	0	0.09	109	0	0.04	152	0	0.05	89	0	0.03
18	22	3000	50	2,837	209	1.22	2,919	195	1.23	2,910	194	1.23	2,303	190	0.77
19	18	3000	50	3,431	487	1.79	3,047	337	1.10	2,922	298	1.03	3,045	298	1.06
19	20	3000	50	143	0	0.05	58	0	0.02	44	0	0.01	47	0	0.02
19	23	3000	50	193	0	0.06	78	0	0.03	129	0	0.04	64	0	0.02
20	19	3000	50	984	24	0.36	1,471	130	0.66	984	54	0.40	1,470	55	0.56
20	17	2500	30	2,980	206	1.52	2,636	166	1.32	2,494	165	0.90	2,389	173	0.88
20	21	3000	50	103	0	0.03	42	0	0.01	32	0	0.01	34	0	0.01
20	24	3000	50	580	23	0.22	682	9	0.24	656	7	0.23	392	8	0.14
21	20	3000	50	261	0	0.09	558	0	0.19	602	0	0.20	958	0	0.32

Note: The links in the shadow are the expanded links.

Table 7.2.3 Network 3, TCNDP Best Solutions and Network Link Flows (Continued Previous Page)

NETWORK 3															
Network Configuration Link		**Capacity**	**Free Flow Speed**	**Initial Nework**			**10% Budget**			**20% Budget**			**30% Budget**		
From	*To*	*(Vehicles)*	*(Miles/Hour)*	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C
21	14	3000	50	3,485	267	1.52	3,225	233	1.39	3,196	241	1.39	3,141	258	1.39
21	25	3000	50	1,690	197	0.83	1,621	225	0.84	1,604	229	0.84	1,819	233	0.92
22	18	3000	50	4,603	801	2.60	3,724	728	1.66	3,623	727	1.63	3,620	705	1.61
22	23	4500	55	4,986	719	1.75	3,727	527	1.06	3,793	513	1.06	3,848	426	1.01
22	26	4500	55	3,150	208	0.89	3,224	195	0.89	3,355	200	0.92	3,482	196	0.95
23	22	4500	55	1,089	133	0.36	620	54	0.19	844	75	0.25	888	81	0.27
23	24	4500	55	2,471	256	0.78	2,394	217	0.72	2,171	176	0.64	2,203	175	0.65
23	19	3000	50	2,515	463	1.46	2,458	310	1.23	2,157	270	0.81	2,392	342	1.25
24	23	4500	55	896	133	0.32	542	54	0.17	863	85	0.27	824	81	0.25
24	20	4500	55	3,738	229	1.03	3,564	296	1.06	3,405	250	0.98	3,411	260	0.99
24	25	4500	55	499	115	0.21	641	102	0.23	677	35	0.18	334	38	0.11
25	24	4500	55	2,083	198	0.64	1,671	226	0.57	2,119	186	0.64	1,973	197	0.61
25	21	4500	55	3,643	267	1.05	3,741	233	1.04	3,767	241	1.05	3,366	213	0.77
25	28	2000	30	2,151	222	1.52	2,076	235	1.51	2,086	236	1.52	1,971	241	1.47
26	22	4500	55	8,813	1,386	3.19	7,735	1,149	2.24	7,452	1,138	2.18	7,313	1,118	2.14
26	27	2000	30	1,520	125	1.01	1,495	103	0.95	1,460	114	0.96	1,571	108	1.00
26	29	2000	55	1,557	138	1.05	1,522	126	1.01	1,160	110	0.57	1,191	93	0.56
27	26	2000	30	4,010	403	2.81	4,071	385	2.81	3,188	292	1.56	3,425	311	2.33
27	28	2000	30	882	49	0.54	914	20	0.50	647	15	0.35	573	16	0.32
27	30	2000	55	1,590	216	1.23	1,517	218	1.19	1,439	142	1.00	1,058	122	0.55
28	25	2000	30	5,687	374	3.59	5,227	366	3.35	4,548	317	2.08	4,700	330	2.15
28	27	2000	30	1,849	231	1.39	2,135	227	1.52	1,953	233	1.44	1,801	221	1.34
28	31	4500	30	2,804	236	0.83	2,411	199	0.58	2,212	202	0.55	2,039	205	0.52
29	26	2000	55	3,530	407	2.58	3,018	316	1.53	2,704	308	1.41	3,042	340	1.57
29	30	2000	30	1,557	138	1.05	1,522	126	1.01	1,523	110	0.98	1,534	113	0.99
30	29	2000	30	3,530	407	2.58	3,795	412	2.72	2,704	308	1.41	3,042	340	1.57
30	27	3500	50	3,113	312	1.25	2,873	294	1.16	2,663	176	0.96	1,533	121	0.45
30	31	4500	30	2,196	214	0.68	2,102	209	0.65	2,336	210	0.71	2,532	205	0.74
31	30	4500	30	5,692	580	1.78	5,731	571	1.78	5,427	529	1.68	5,419	516	1.66
31	28	4500	30	7,308	570	2.13	6,025	480	1.44	6,279	509	1.51	6,281	513	1.52

Note: The links in the shadow are the expanded links.

Table 7.2.4 Network 4, TCNDP Solutions and Network Link Flows (V/C ratio > 0.1 Only)

NETWORK 4															
Network Configuration Link		Capacity	Free Flow Speed	Initial Nework			10% Budget			20% Budget			30% Budget		
From	To	(Vehicles)	(Miles/Hour)	Car Flow (Vehicles)	Truck Flow (Vehicles)	V/C	Car Flow (Vehicles)	Truck Flow (Vehicles)	V/C	Car Flow (Vehicles)	Truck Flow (Vehicles)	V/C	Car Flow (Vehicles)	Truck Flow (Vehicles)	V/C
23	28	1500	30	1,379	198	1.45	1,370	134	1.27	1,179	120	0.72	1,129	123	0.70
24	25	1500	30	482	0	0.32	189	0	0.13	165	0	0.11	147	0	0.10
24	30	4500	55	3,955	436	1.27	3,908	414	1.24	3,268	341	0.81	3,341	344	0.83
24	29	1500	30	1,744	137	1.53	1,532	148	0.92	1,366	137	0.83	1,353	135	0.82
24	23	1500	30	1,379	198	1.45	1,370	134	1.27	1,179	120	0.72	1,129	123	0.70
25	30	1500	30	576	0	0.38	478	0	0.32	198	0	0.13	176	0	0.12
26	31	1500	30	1,593	0	1.06	1,126	0	0.75	800	0	0.53	908	0	0.39
27	26	1500	30	1,747	20	1.22	1,438	8	0.98	852	7	0.59	929	6	0.41
27	32	1500	30	151	5	0.11	250	2	0.17	76	2	0.06	153	1	0.11
27	31	4500	55	4,556	804	1.73	4,592	723	1.66	4,420	585	1.19	4,356	658	1.23
28	33	2500	45	2,667	301	1.55	2,566	242	1.41	2,495	236	0.98	2,522	231	0.98
29	28	1500	30	1,289	103	1.13	1,196	108	1.08	918	73	0.81	760	51	0.42
29	34	2500	45	1,574	108	0.80	1,603	133	0.85	1,635	135	0.87	1,442	137	0.57
30	29	1500	30	1,119	62	0.91	743	27	0.57	727	23	0.54	627	19	0.47
30	31	4500	55	3,000	300	0.93	3,000	300	0.93	3,000	300	0.74	2,383	300	0.63
30	35	4500	55	4,289	391	1.30	3,821	532	1.32	3,816	515	1.03	3,669	464	0.97
31	30	4500	55	3,877	318	1.14	3,178	445	1.10	2,824	351	0.74	3,160	390	0.83
31	44	4500	55	508	45	0.15	413	48	0.13	478	48	0.15	396	49	0.13
31	36	1500	30	448	224	0.90	519	88	0.58	512	77	0.55	470	69	0.50
32	31	1500	30	2,281	253	2.20	2,181	218	1.33	2,164	210	1.31	2,019	214	1.25
32	27	1500	30	128	0	0.09	50	0	0.03	205	0	0.14	39	0	0.03
32	44	2500	35	1,642	35	0.71	2,082	42	0.90	1,388	22	0.59	1,282	11	0.53
34	33	1500	30	1,416	129	1.29	1,346	158	1.32	1,103	147	0.73	971	136	0.66
34	35	4500	55	1,674	32	0.40	1,576	13	0.36	854	11	0.20	518	10	0.12
34	39	4500	55	3,176	179	0.86	3,103	208	0.87	2,676	180	0.75	2,719	208	0.79
35	34	4500	55	1,811	141	0.53	1,246	130	0.39	768	100	0.26	1,007	131	0.34
36	35	1500	30	1,149	248	1.43	1,149	116	1.07	1,012	87	0.59	1,022	76	0.58
36	31	1500	30	467	13	0.35	320	5	0.23	270	5	0.19	197	4	0.14
37	32	1500	30	3,900	284	3.36	3,612	225	1.96	3,186	230	1.79	2,901	220	1.64
37	27	2500	35	3,025	498	2.01	2,929	403	1.30	2,717	389	1.22	2,597	365	1.16
37	46	2500	35	3,218	291	1.75	3,024	346	1.26	2,532	261	1.02	2,470	211	0.95
37	44	1500	30	2,904	241	2.58	2,422	198	1.40	2,131	193	1.26	2,021	196	1.22
38	33	2500	45	1,094	129	0.64	1,158	124	0.66	993	138	0.62	1,266	106	0.68
39	38	1500	30	853	129	0.91	1,063	124	1.04	911	124	0.94	889	80	0.53

Note: The links in the shadow are the expanded links.

Table 7.2.4 Network 4, TCNDP Solutions and Network Link Flows (V/C ratio > 0.1 Only) **(Continued Previous Page)**

NETWORK 4															
Network Configuration Link		**Capacity**	**Free Flow Speed**	**Initial Nework**			**10% Budget**			**20% Budget**			**30% Budget**		
From	*To*	*(Vehicles)*	*(Miles/Hour)*	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C
39	48	2500	45	6,479	569	3.50	6,411	590	2.51	6,272	589	2.47	6,105	590	2.42
40	39	1500	30	1,412	194	1.46	1,457	176	1.44	1,371	166	0.89	1,386	167	0.89
41	40	1500	30	1,186	124	1.12	1,206	115	1.11	1,000	131	1.01	723	133	0.55
41	43	1500	30	1,276	70	1.04	1,258	92	1.08	1,063	94	0.63	932	60	0.51
41	51	1500	30	239	51	0.30	184	20	0.18	184	18	0.17	142	16	0.14
42	34	2500	45	2,678	57	1.16	3,097	101	1.40	2,551	90	0.83	2,025	88	0.95
42	40	1500	30	415	69	0.46	459	61	0.31	882	98	0.85	728	63	0.43
43	36	1500	30	1,168	37	0.88	950	33	0.72	1,156	32	0.86	808	26	0.40
43	42	2500	45	2,667	126	1.27	3,289	163	1.13	3,228	189	1.14	2,845	175	1.01
44	43	2500	35	2,754	94	1.25	2,421	57	0.76	2,256	126	0.79	2,181	79	0.71
44	31	4500	55	4,437	367	1.31	3,722	382	1.17	3,697	386	1.16	3,183	288	0.76
44	54	4500	55	3,789	453	1.24	3,402	417	1.13	3,341	405	1.10	3,205	404	0.85
45	41	1500	30	2,234	229	2.10	1,643	164	1.00	1,668	177	1.03	1,643	169	1.01
45	44	1500	30	2,314	230	2.16	1,708	191	1.07	1,803	187	1.11	1,806	182	1.10
46	44	1500	30	2,886	260	2.62	2,035	220	1.27	2,105	216	1.29	2,100	206	1.27
46	47	2500	35	1,266	72	0.62	832	72	0.45	995	75	0.52	942	58	0.47
47	45	1500	30	2,149	246	2.09	1,565	176	1.51	1,656	183	1.04	1,622	169	1.00
47	46	2500	35	980	55	0.48	542	51	0.30	534	46	0.29	593	65	0.34
47	55	2500	35	1,953	101	0.94	1,618	127	0.85	1,734	115	0.88	1,641	97	0.81
49	48	1500	30	1,085	122	1.05	1,115	137	1.11	940	115	0.61	1,029	106	0.63
50	49	1500	30	1,085	122	1.05	1,115	137	1.11	940	115	0.61	1,029	106	0.63
51	50	1500	30	1,322	122	1.21	1,173	137	1.15	1,030	115	0.65	1,021	106	0.63
51	41	1500	30	467	17	0.36	445	7	0.31	430	6	0.30	245	5	0.18
51	54	1500	30	1,329	152	1.29	1,322	144	1.27	1,142	128	1.10	976	92	0.58
52	51	1500	30	2,879	240	2.56	2,043	200	1.24	2,083	200	1.25	2,081	189	1.23
52	45	1500	30	2,398	213	2.17	1,745	189	1.09	1,832	181	1.11	1,823	181	1.11
52	55	1500	30	1,489	265	1.70	1,038	156	0.72	940	160	0.69	1,006	132	0.67
54	39	4500	55	2,945	372	0.98	2,993	347	0.97	2,838	321	0.92	2,777	315	0.90
55	54	1500	30	2,659	299	2.57	2,094	192	1.24	2,131	186	1.25	2,057	186	1.22
55	52	1500	30	265	68	0.36	104	27	0.14	91	23	0.12	81	21	0.11
55	47	2500	35	517	0	0.21	203	0	0.08	178	0	0.07	158	0	0.06

Note: The links in the shadow are the expanded links.

Table 7.2.5 Network 5, TCNDP Solutions and Network Link Flows (V/C ratio > 1.0 Only)

NETWORK 5															
Network Configuration Link		Capacity	Free Flow Speed	Initial Nework			10% Budget			20% Budget			30% Budget		
From	To	(Vehicles)	(Miles/Hour)	Car Flow (Vehicles)	Truck Flow (Vehicles)	V/C	Car Flow (Vehicles)	Truck Flow (Vehicles)	V/C	Car Flow (Vehicles)	Truck Flow (Vehicles)	V/C	Car Flow (Vehicles)	Truck Flow (Vehicles)	V/C
13	14	500	30	903	215	3.52	728	163	1.38	491	120	0.97	500	118	0.97
13	19	500	30	897	235	3.68	585	144	1.16	409	105	0.83	449	117	0.92
14	13	500	30	979	29	2.19	623	42	0.79	636	23	0.73	674	27	0.78
14	15	500	30	726	136	2.54	618	117	1.08	473	102	0.88	461	98	0.85
14	20	500	30	599	114	2.11	338	56	1.12	340	44	1.03	334	56	1.12
15	16	500	30	534	112	1.96	473	117	1.88	417	104	1.67	367	90	1.45
16	15	500	30	364	69	1.28	447	67	1.43	351	67	1.24	354	58	1.17
16	17	500	30	394	66	1.31	392	63	1.29	366	86	1.42	324	74	1.24
17	16	500	30	708	164	2.73	605	127	1.11	445	111	0.89	400	86	0.74
17	18	500	30	1,018	80	2.67	553	94	1.86	694	66	0.96	666	69	0.94
17	23	500	30	685	177	2.79	390	136	1.87	380	135	1.84	293	91	0.66
18	17	500	30	1,011	214	3.73	656	142	2.45	498	119	0.97	502	122	0.99
18	24	500	30	789	256	3.63	571	164	1.23	451	144	1.03	435	137	0.98
19	13	500	30	821	31	1.89	576	18	1.30	531	13	1.17	575	12	1.24
19	20	500	30	541	132	2.14	593	146	2.35	412	104	1.66	356	81	1.36
19	25	500	30	356	105	1.55	582	142	2.30	400	107	1.66	292	82	0.62
20	21	500	30	563	122	2.10	567	119	2.08	419	97	1.61	400	94	1.55
21	22	500	30	488	127	1.99	468	117	1.87	411	108	0.84	373	88	0.73
22	21	500	30	470	62	1.43	441	67	1.41	305	61	1.10	353	55	1.15
22	57	700	40	1,034	73	1.89	912	100	1.87	795	120	1.06	846	76	0.96
23	22	500	30	546	126	2.10	590	77	1.80	400	75	1.40	314	52	0.52
23	24	500	30	462	171	2.29	320	81	1.29	340	104	1.51	403	74	1.40
23	29	500	30	435	212	2.57	285	145	1.73	278	138	1.66	298	121	1.56
24	18	500	30	782	100	2.37	509	86	1.71	586	61	1.66	588	70	1.73
24	23	500	30	492	132	2.04	525	92	1.78	383	77	1.38	454	58	1.37
24	30	500	30	297	246	2.56	300	151	0.90	264	149	0.86	225	138	0.78
25	26	500	30	391	72	1.36	413	106	1.67	306	59	1.08	300	56	1.05
25	31	500	30	322	77	1.26	495	125	1.99	319	83	1.30	275	76	1.16
26	27	500	30	470	108	1.80	508	119	1.96	409	87	1.51	386	84	1.45
27	26	500	30	369	30	0.98	334	76	1.27	260	48	0.90	302	63	1.10
27	28	500	30	409	130	1.86	466	119	1.89	370	118	1.68	336	101	1.48
28	27	500	30	532	53	1.49	459	57	1.37	379	33	1.02	391	43	1.12
28	29	500	30	336	269	2.82	89	157	1.44	113	164	0.77	103	134	0.64
29	30	500	30	381	119	1.72	202	146	1.57	206	142	1.54	261	104	1.35

Note: The links in the shadow are the expanded links.

Table 7.2.5 Network 5, TCNDP Solutions and Network Link Flows (V/C ratio > 1.0 Only) (Continued Previous Page)

NETWORK 5															
Network Configuration Link		Capacity	Free Flow Speed	Initial Nework			10% Budget			20% Budget			30% Budget		
				Car Flow	Truck Flow	V/C	Car Flow	Truck Flow	V/C	Car Flow	Truck Flow	V/C	Car Flow	Truck Flow	V/C
From	*To*	*(Vehicles)*	*(Miles/Hour)*	*(Vehicles)*	*(Vehicles)*		*(Vehicles)*	*(Vehicles)*		*(Vehicles)*	*(Vehicles)*		*(Vehicles)*	*(Vehicles)*	
29	35	500	30	262	346	3.29	167	170	1.69	113	165	0.77	75	152	0.68
30	36	500	30	309	343	3.36	305	182	1.04	189	207	1.02	240	166	0.91
31	37	500	30	725	180	2.89	504	121	0.99	575	130	1.10	436	106	0.86
32	31	500	30	402	103	1.63	510	117	1.96	257	54	0.95	324	58	1.11
32	33	500	30	436	104	1.70	480	118	1.90	451	99	1.69	406	88	1.51
32	38	500	30	349	96	1.47	275	69	1.10	386	102	1.59	274	85	1.23
33	32	500	30	397	88	1.50	410	88	1.52	344	77	1.31	359	78	1.34
33	34	500	30	404	119	1.76	426	133	1.91	379	107	1.61	290	98	0.68
34	33	500	30	486	89	1.69	486	60	1.45	436	47	1.25	465	52	1.34
34	35	500	30	498	284	3.26	242	180	0.96	180	193	0.95	169	170	0.85
34	59	700	40	834	28	1.35	676	18	1.07	565	25	0.95	547	39	1.00
35	34	500	30	475	58	1.41	590	52	1.59	493	33	1.25	343	63	0.59
35	36	500	30	689	308	3.84	377	159	1.01	263	200	1.07	282	158	0.91
35	41	500	30	107	322	2.79	123	218	1.99	64	157	0.69	176	139	0.73
36	42	500	30	989	650	7.18	366	352	1.77	429	404	2.05	399	337	1.75
37	31	500	30	735	153	2.69	689	146	2.54	470	85	1.62	436	76	1.48
37	38	500	30	701	159	2.67	745	152	2.71	457	87	0.80	410	90	0.77
37	43	500	30	765	138	2.64	765	153	2.75	407	96	0.79	472	98	0.87
38	37	500	30	768	167	2.87	623	132	2.30	540	123	1.03	438	99	0.83
38	39	500	30	508	117	1.95	553	110	1.99	566	98	1.92	505	87	1.71
39	38	500	30	458	98	1.70	444	92	1.63	479	105	1.80	464	90	1.65
39	40	500	30	511	252	3.04	315	145	1.79	250	122	1.47	190	106	1.23
39	51	700	40	936	95	1.88	805	76	1.58	737	83	1.53	727	49	0.77
40	39	500	30	501	74	1.59	567	68	1.68	548	69	1.64	495	67	0.76
40	41	500	30	434	345	3.63	190	233	1.12	321	191	1.08	272	182	1.00
41	40	500	30	601	79	1.84	542	69	1.63	462	59	1.40	485	81	1.62
41	42	500	30	661	702	6.94	392	335	1.73	454	409	2.09	393	345	1.77
42	36	500	30	731	100	2.26	702	99	2.19	562	66	1.65	440	55	0.66
42	41	500	30	762	98	2.31	801	99	2.39	596	64	0.85	440	57	0.67
42	46	500	30	707	102	2.23	698	102	2.21	513	76	1.63	438	76	1.48
43	37	500	30	707	153	2.64	573	125	2.15	522	113	1.95	448	94	1.65
43	44	500	30	540	101	1.89	531	101	1.87	511	103	1.85	415	75	0.71
43	47	500	30	357	103	1.54	341	107	1.54	341	109	1.55	156	92	1.05
44	43	500	30	439	84	1.55	326	80	1.29	437	90	1.60	318	66	1.17

Note: The links in the shadow are the expanded links.

Table 7.2.5 Network 5 TCNDP Solutions and Network Link Flows (V/C ratio > 1.0 Only) (Continued Previous Page)

NETWORK 5															
Network Configuration Link		Capacity	Free Flow Speed	Initial Nework			10% Budget			20% Budget			30% Budget		
From	*To*	*(Vehicles)*	*(Miles/Hour)*	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C	Car Flow *(Vehicles)*	Truck Flow *(Vehicles)*	V/C
44	45	500	30	209	117	1.35	241	157	1.74	214	110	1.31	163	113	1.23
44	50	500	30	849	46	2.06	715	38	0.87	717	32	0.85	572	38	0.73
45	44	500	30	554	62	1.60	587	59	1.64	524	54	0.74	518	65	1.55
45	46	500	30	404	346	3.57	231	228	1.15	180	176	0.89	269	166	0.93
46	42	500	30	549	648	6.28	341	317	1.61	418	384	1.95	321	320	1.60
47	48	500	30	160	112	1.22	223	157	1.71	124	120	1.21	108	132	1.27
48	46	500	30	293	337	3.28	203	213	2.11	130	153	1.49	76	169	1.51
48	47	500	30	473	64	1.46	482	63	1.47	400	60	1.28	379	53	1.18
49	62	5250	55	6,300	1,120	2.05	6,300	1,123	2.06	4,787	783	1.51	4,768	820	1.25
50	49	5250	55	6,300	630	1.68	6,300	633	1.68	6,300	630	1.68	4,840	413	1.01
51	50	5250	55	5,451	584	1.48	4,662	549	1.31	4,683	550	1.31	3,590	559	0.90
52	51	5250	55	4,515	573	1.30	3,936	604	1.21	3,945	543	1.17	3,727	538	1.12
53	33	700	40	378	41	0.77	700	60	1.34	283	67	0.79	518	67	1.13
53	52	5250	55	3,996	661	1.26	3,517	652	1.17	3,595	581	1.13	3,396	577	1.09
54	53	5250	55	3,671	633	1.18	3,697	677	1.22	3,490	611	1.13	3,469	618	1.13
55	21	700	40	625	167	1.85	477	73	1.10	253	62	0.71	507	70	1.12
55	54	5250	55	4,059	642	1.26	3,834	702	1.27	3,528	621	1.15	3,661	646	1.19
56	14	1400	40	1,221	57	1.03	1,204	45	0.99	1,204	45	0.99	1,237	74	1.10
56	17	1400	40	985	132	1.08	1,088	140	1.18	1,120	196	1.36	1,213	181	1.38
56	55	5250	55	4,414	728	1.40	4,240	753	1.38	4,265	677	1.33	4,103	670	1.29
57	56	5250	55	6,620	337	1.52	6,532	359	1.52	5,021	215	0.91	4,994	210	0.90
58	57	5250	55	5,736	283	1.31	5,659	266	1.28	4,800	225	1.09	4,891	202	1.09
61	40	700	40	969	181	2.42	755	217	2.32	454	143	1.47	619	134	1.65
61	60	5250	55	4,924	662	1.44	4,753	695	1.43	3,970	513	1.15	3,792	521	1.12
62	45	700	40	722	300	2.75	875	216	2.49	693	178	1.17	683	169	1.13
62	61	5250	55	5,578	820	1.69	5,425	906	1.72	4,706	780	1.49	3,747	554	0.92

Note: The links in the shadow are the expanded links.

3) Most of the links with high V/C ratios are included for link expansion, but some of the links of with smaller V/C ratios were also selected for expansion. This observation further demonstrates that the V/C ratio provides a valuable information that can be used as one of the criteria for link selection. One of the most important features of the network design problem is that every single link in a network is not an isolated individual link but it is a member for a number of paths, and the selection of the link to be expanded relies more on the path characteristics. Thus a low V/C ratio for a link which is a part of a vital congested path can improve the network performance more than an otherwise isolated link (which is a member of only a few paths) with a high V/C ratio.

CHAPTER 8

CONCLUSION AND FUTURE RESEARCH

8.1 Summary

The motivation for this study stems from the recognition that the traffic stream may be divided into two distinct classes of users with different operational characteristics, passenger cars and trucks. Various states have designated specific truck routes, climbing lanes are widely used throughout the country, and trucks are prohibited from entering the left lane of almost all major highways. In addition, several states are considering the implementation of special truck lanes, to improve the safety and operational efficiency of roadways with high truck volumes.

The specific problem addressed in this dissertation is addressed only the operational aspect of truck lane needs which is outlined below:

Given the passenger and truck origin-destination (O-D) matrices, the available budget and a highway network, identify the best combination of the network links for capacity expansion and operational improvements, in minimizing the total network UE travel time. The following link improvements were considered:

1. Do not expand link,
2. Both passenger cars and trucks allowed on new lane,
3. Only passenger cars allowed on new lane,
4. Only trucks allowed on new lane.

The above problem is classified as a mixed integer non-linear problem. Its complexity falls under the category of Non-polynomially hard problems. In the worst case it requires 4^n (n is the total number of candidate links) iterations to enumerate all possible solution states of the problem. Even for very small networks the computational time required to solve the problem is prohibitive (e.g. assuming a network with 10 candidate links, the total number of iterations would be 1,048,576). Therefore, branch and bound based procedures cannot be used to efficiently solve even moderately small size problems.

The major contribution of this dissertation is the development of a methodology in providing "good" solutions to the above problem. The methodology developed utilizes a combination of simulated annealing and tabu search strategies. These strategies have been widely used in solving large scale combinatorial problems. While it is recognized that these procedures may not reach a global optimum, for large scale problems near-optimal solutions may be sufficient.

The combined simulated annealing and tabu search strategy (SA-TABU) developed requires at every iteration to solve a multi-class (passenger cars and trucks) traffic assignment procedure with asymmetric link interactions, to identify the flows on the links of the network. The diagonalization algorithm was used to solve the traffic assignment at every iteration of the SA-TABU search strategy developed. In all iterations, the traffic assignment procedure converged, which implies that equilibrium was reached at each iteration. The solution to the traffic assignment problem is also the major burden to the computational time required by the search strategy, especially for large networks.

The specific travel cost functions used in the traffic assignment, were derived primarily from engineering judgment using a modification of the BPR type curves and the concept of passenger car equivalents as applied by the 1994 HCM. In particular the travel cost functions were set in such a way in order to reflect the higher contribution that the cars place on passenger cars' travel time and the lesser contribution of the passenger cars on the trucks' travel time. The Jacobian of these travel cost functions is proven to be positive definite which guarantees that the UE solution found is also unique.

The SA-TABU search strategy developed combines the advantage of the conventional simulated annealing procedure to guide the search in a systematic way, the tabu search strategy's advantage in reducing the risk of cycling and in avoiding local optima, and the characteristics of the two classes of users network design problem. A major contribution in this study is the development of a comprehensive heuristic evaluation function (HEF) which is used to evaluate each available move at every iteration which is composed of three elements: the link's volume to capacity ratio which captures the current flow characteristics of the link, the historical contribution of the link to the objective function that is updated continuously throughout the search (LCOF), and a random variable which provides a stochastic nature to the HEF developed.

Table 8.2.1 Summary of Selected Results

Network Improvement and Budget

Network	Budget Level 10%	20%	30%
1	73.3%	74.3%	76.7%
2	32.2%	82.4%	85.4%
3	94.5%	98.9%	99.2%
4	84.7%	93.6%	98.4%
5	97.0%	98.5%	99.3%

SA ,SA-TABU Comparison-"Best Solution" & "Best Solution" 1st Appearance( )

	10%		20%		30%	
Network	SA	SA-TABU	SA	SA-TABU	SA	SA-TABU
1	7541(980)	6605(656)	7296(1001)	6488(544)	6232(1756)	6232(103)
2	2232(1333)	2171(467)	1876(1578)	1720(1167)	1765(1602)	1693(1426)
3	40213(1899)	38584(1164)	37000(1772)	37024(846)	37001(1926)	36920(1225)
4	10391(1265)	10520(554)	10099(1993)	9756(1564)	9698(901)	9343(1154)
5	24483(1820)	23512(738)	22019(1537)	20005(1656)	19973(1997)	18269(1764)

Passenger Cars and Trucks Total Travel Time Reduction

Network	Passenger Cars	Trucks
1	74.1%	74.5%
2	38.5%	3.5%
3	49.1%	53.8%
4	48.4%	47.9%
5	95.2%	76.6%

Break Point of Exclusive Car Lane and Exclusive Truck Lane

	Overall V/C ratio 0.10	0.5	0.875	1	1.5
Truck Flow / Paasenger Car Flow	70%	33%	35%	35%	37%

8.2 Conclusions

The following findings and conclusions are abstracted from the summary and results of the analysis of the numerical experiments presented in Chapter 7 and Chapter 8.

8.2.1 TCNDP Characteristics

1) Link Selection for Expansion

- The probability that the links selected for capacity expansion, have higher volume to capacity ratio than the links not selected for capacity expansion, is very high. Links selected for capacity expansion with relatively low volume have also been observed to be included in the optimal solution. These links are either members of a critical path or they are included due to the budget constraint which may not allow links with higher volume to capacity ratio but having a higher cost to enter the solution.
- The higher the budget level the larger its contribution is to the network total travel time. However, usually most the contribution to the total travel time of the network is achieved with low budget levels. This can be clearly observed from "Network improvement and Budget" in Table 8.2.1.
- The links selected for expansion with lower budget levels are not necessarily included at the solutions with higher budget levels. This result exemplifies the user behavior in selecting their routes based on the current network configuration.
- Table 8.2.1 indicates that under the situation where all the existing links were allowed to be expanded by one shared lane, for both passenger cars and trucks for

the five test networks, the passenger cars experience more travel time reduction than the trucks in most of the cases.

2) Lane Traffic Operation Selections

- In most of the tests conducted the selected link expansion option was the shared lane with both passenger cars and trucks.
- In limited cases, the exclusive lane is either a truck or a passenger car lane, which occurs when one class dominates the other class. In Table 8.2.1 under the heading "Break Point of Exclusive Car Lane and Exclusive Truck Lane", the break point of selecting exclusive truck lane or passenger car lane various congestion level (overall V/C ratio) is presented. With moderate to heavy congestion levels, the exclusive passenger car lane would be the better selection than the exclusive truck lane when the ratio of truck flow to car flow is lower than around 33% in the experiment. However, the lane expansion for both passenger cars and trucks is selected most of the times rather than the lane expansion for exclusive use.
- If the network is not congested, there is no significant difference between the three different options. When the network becomes more congested, the differences between the different options are much more distinct.

3) Traffic Flow

- The expanded links usually attract more traffic than the level of traffic they experienced prior to the expansion. This result applies primarily to the expanded links with high volume to capacity ratios.

- The expanded link's volume to capacity ratio usually decreases. However, in some occasions it may increase by attracting more traffic.
- The mitigation of the congestion usually leads to the reduction of the total travel times. However, in some occasions while the total travel time is reduced for some links their corresponding v/c ratio increases.

4) Heuristic Evaluation Function (HEF)

- The traffic flow parameters provide important information (e.g. V/C ratio) that may be used for the selection of link candidates for expansion.
- The combined passenger car and truck volume to capacity ratio provides a very valuable parameter for the selection of the links to be expanded, especially when the passenger car and truck volumes are relatively close to each other.
- The individual passenger cars or trucks volume to capacity ratios are not as valuable, especially when the links are not congested.
- The link's contribution in reducing the total travel time is also an important component of the heuristic evaluation function, and it becomes quite crucial when the information based on the links attributes are difficult to be identified.
- The random error terms expands the search space and reduces the risk of cycling.

8.2.2 Characteristics of the SA-TABU Search Strategy and the Traffic Assignment

The primary conclusions regarding the performance of the SA-TABU search strategy are summarized below:

- The SA-TABU search strategy is an efficient and robust algorithm in providing "good" solutions to the TCNDP.
- In comparison to the conventional SA algorithm, the SA-TABU directs its search faster towards a set of "good' solutions, that makes it applicable for the solution of large scale problems (network sizes). In contrast, SA becomes inefficient for large scale problems due to the explanation of a much larger set of solutions. Table 8.2.1 under the heading " SA, SA-TABU Comparison-'Best Solution' and First Appearance of 'Best Solution' " shows the performance difference is accelerated by the increase in network size, when measured by the "best solution" generated.
- Table 8.2.1 " SA, SA-TABU Comparison-'Best Solution' and First Appearance of 'Best Solution' " also presents the iteration when the "best solution" first appears (the number is in bracket). The SA-TABU found a better "best solution" much earlier than SA.
- The most important components of the algorithm, such as the Markov chain length, "cooling schedule", tabu length and heuristics function, can be customized for different problems and objectives.
- The Markov chain length and "cooling schedule" determine the search process's mechanic features. The longer the Markov chain length is and the smaller the control parameter or "temperature" dropping rate is, the better the final solution is, while the longer time is required, and vice versa.
- Network 1 and Network 2 are relatively small size network, which are favored by the longer Markov chain length and the slower "temperature" dropping rate such

as the Version 1 SA-TABU search strategy, and Networks 3,4, and 5 would prefer Version 2 SA-TABU which features a shorter Markov chain length and a faster "temperature" dropping rate. The shorter the tabu list length SA-TABU in Version 3 performs well for the small networks, Network 1 and Network 2. The version with a heavy weight on the V/C ratio information in the HEF such as the Version 6 SA-TABU, exhibit an advantage in the numerical experiments for the larger test networks, Networks 3,4, and 5.

- If the link attributes such as the V/C ratio are found to be informative then the weight of this attribute in the HEF can be greater, otherwise the weight of the random variable or the link historical contribution should be set higher.
- The tabu list length is determined by the number of feasible solutions which is determined by the network size and the budget level.
- The use of the modified link travel cost function for the two classes of users guarantees that the traffic assignment converges to a unique equilibrium solution since its Jacobian is positive definite.

8.3 Future Research

In this study, the primary objective was the reduction of the total travel time of the network. However, other important objectives may also be considered in determining truck routes and truck lane needs such as, safety, roadway pavement life time, environmental impacts. In addition, the passenger cars and trucks travel times may be considered as two separate objectives in the formulation. The consideration of these objectives in the TCNDP would be very useful and more realistic to the engineering practices. The problem can be set as a multi-objective one, minimizing the total travel time of the network, maximizing safety, minimizing the truck travel time etc..

In this study, the modified passenger car and truck link cost functions were used in order to ensure that the diagonalization algorithm converges to an equilibrium flow in the two classes of users traffic assignment. Future research is needed in developing actual travel cost functions based on a comprehensive traffic flow and travel time data at various transportation facilities.

The SA-TABU search strategy performed very well for the solution to the TCNDP. Other techniques may be considered which can further improve the performance of the search procedure. In particular, other advancements in combinatorial optimization such as the utilization of elements from neural networks, genetic algorithms may be considered.

Develop a set of benchmark problems for the TCNDP that may be used to continuously compare several algorithms or search strategies. In most of the heuristic search strategies the global optimum solution is not guaranteed, however, the benchmark problems will aid in the comparison on the quality of the solutions obtained.

The heuristic evaluation function plays an important role in determining the efficiency and effectiveness of the search strategy. Though the HEF developed in this study is quite robust, it doesn't take full advantage of the network design characteristics. Efforts should be undertaken to focus more on the passenger car and truck respective congestion index and more important, the path link structure. The traffic assignment is primarily based on path flows rather than link flows. Therefore, by identifying the paths that pass through critical links more insights can be found in identifying the critical links of the network to be expanded. In addition, other elements may be considered to become parts of the HEF, as well as to consider other forms not necessarily linear. A sensitivity analysis for various types of HEF would be beneficial in identifying the best form of the HEF.

The search strategy currently utilized the add/drop type of move, by dropping one element and adding as many as to satisfy the budget constraint. An alternate procedure may be sought where two or more elements are dropped and an appropriate number of links are then added to satisfy the budget constraint.

Another area of research is to try and optimize the computational efficiency of the search strategy through parallel computing. Several aspects of the search strategy can be optimized such as the traffic assignment procedure, and the heuristic search itself.

REFERENCES

E. Aarts and J. Korst, *Simulated Annealing and Boltzman Machines.* Interscience Series Discrete Mathematics and Optimization, John Wiley & Sons, Inc., New York, New York, 1989.

M. Abdulaal and L. J. LeBlanc, "Continuous Equilibrium Network Design Models," *Transportation Research*, Volume 13B, Number 1, pp. 19-33, March 1979.

M. D. Amico and Marco Trubian, "Applying tabu search to the job-shop scheduling problem," *Annuals of Operations Research*, 41, pp. 231-252, 1993.

Y. Arezki and D. V. Vliet, "A Full Analytical Implementation of the PARTAN/Frank-Wolfe Algorithm for Equilibrium Assignment," *Transportation Science*, Volume 24, Number 1, February 1990.

A. Balakrishnan, T. L. Magnanti, and R.T. Wong, "A Dual - Ascent Procedure For Large Scale Uncapacitated Network Design," *Operational Research*, Vol. 37, No.5, September-October 1989.

J. Bard, "An Algorithm for Solving the General Bilevel Programming Problem," *Mathematics of Operation Research*, 8, pp. 260-272, 1983.

O. Ben-Ayed, D. E. Boyce, and C. E. Blair III, "A General Bilevel Linear Programming Formulation of the Network Design Problem," *Transportation Research Board*, Volume 22B, No. 4, pp. 311-318, 1988.

S. Berka, J. Raj, and A. Tarko, "Turning Specific Link Travel Time Functions for Network Modeling," *Advance Working Paper Series Number 26, University of Illinois at Chicago*, July 1993.

D. E. Boyce and J. L. Soberanes, "Solutions to the Optimal Network Design Problem with Shipments Related to Transportation Cost," *Transportation Research Board*, Volume 13B, pp. 65-80, 1979.

S. Catoni and S. Pallottino, "Traffic Equilibrium Paradoxes," *Transportation Science*. Volume 25, No.3, pp. 240-244, August 1991.

Y. Chan, "Transportation Network Investment Problem - A Synthesis of Tree-Search Algorithms," *Transportation Research Record*,1074, 1986.

Y. Chan, T. S. Shen, and N. M. Mahaba, "Transportation-Network Design Problem: Application of a Hierarchical Search Algorithm," *Transportation Research Record*, 1251, 1989.

M. Chen and A. S. Alfa, "A Network Design Algorithm Using a Stochastic Incremental Traffic Assignment Approach," *Transportation Science*, Volume 25, No. 3, August 1991.

G. B. Dantzig, R. P. Harvey, Z. F. Lansdowne, D. W. Robinson, and S. F. Maier, "Formulating and Solving the Network Design Problem by Decomposition," *Transportation Research Board*, Vol.13B, pp. 5-17, 1979.

C. S. Fisk, "Effects of Heavy Traffic on Network Congestion," *Transportation Research Board*. Vol. 24B, No.5, pp. 391-404, 1990.

T. L. Friesz, "Transportation Network Equilibrium, Design and Aggregation: Key Developments and Research Opportunities," *Transportation Research*, Volume 19A, No. 5/6, pp. 413-427, 1985.

T. L. Friesz, H. Cho, N. J. Mehta, R. L. Tobin, and G. Anandalingam, "A Simulated Annealing Approach to the Network Design Problem with Variational Inequality Constraints," Working Paper, July, 1990.

F. Glover, "Tabu Search - Part I," *ORSA Journal on Computing*, Vol. 1, No.3, Summer 1989.

F. Glover, "Tabu Search - Part II," *ORSA Journal on Computing*, Vol. 2, No.3, Winter 1989.

A. E. Haghani and M. S. Daskin, "Network Design Application of an Extraction Algorithm for Network Aggregation," *Transportation Research Record*, 994, pp. 37-46, 1984.

D.W. Hearn, "Fundamental Research on Transportation Networks: Workshop Report," *Transportation Research*, Volume 19A, No. 5/6, pp. 411-412, 1985.

J. E. Hicks, D. E. Boyce, and A. Sen, "Static Network Equilibrium Models and analyses for the Design of Dynamic Route Guidance Systems," *Illinois Universities Transportation Research Consortium*, 1992.

J. N. Hooker, "Needed: an Empirical Science of Algorithms," *Operations Research*, Volume 42, No.2, March-April 1994.

R. Jayakrishnan, K. Wei, Prashker, J. N. Tsai, and S. Rajadhyaksha, "A Faster Path-Based Algorithm For Traffic Assignment," Present in *INFORMS Conference*, Washington, D. C., Spring 1996.

J. S. Kapov, "Tabu Search Applied to the Quadratic Assignment Problem," *ORSA Journal on Computing*, Volume 2, No.1, Winter 1990.

S. Kawakami, H. Lu, and Y. Hirobata, "Estimation of Origin-Destination Matrices From Link traffic Counts Considering the Interaction of the Traffic Modes," *The Journal of the RSAI*, 71,2, pp.139-51, 1995.

J. Knox, and F. Glover, "Tabu Search, An Effective Heuristic for Combinatorial Optimization Problems," Center for Applied Artificial Intelligence, University of Colorado, March 1988.

M. Laguna, J. W. Barnes and F. Glover, "Tabu Search Methods for a Singular Machine Scheduling Problem," *Journal of Intelligent Manufacturing*, Vol. 2, pp. 63-73, 1991.

L. J. LeBlanc, "An algorithm for the Discrete Network Design Problem," *Transportation Science*, pp. 183-199, 5/24/1987.

L. J. LeBlanc and D. E. Boyce, "A Bilevel Programming Algorithm for Exact Solution of the Network Design Problem with User-Optimal Flows," *Transportation Research Board*, Vol. 20B, No.3, pp. 259-265, 1986.

L. J. LeBlanc and M. Abdulaal, "A Comparison of User-Optimum Versus System-Optimum Traffic Assignment In Transportation Network Design," *Transportation Research*, Volume 18B, No.2, pp.115-121, 1984.

C. Lee, "Network Design of One Way Streets: A Mathematical Programming Approach," Working Paper for the Consideration of 1994 TRB Presentation, 1994.

C. Lee, L. Lei, and M. Pinedo, "Recent Development in Search Algorithms Applied to Scheduling," Working Paper, Department of Industrial Engineering, Columbia University, 1995.

M. Los and C. Lardinois, "Combinatorial Programming, Statistical Optimization and the Optimal Transportation Network Problem," *Transporiation Research Board*, Vol.16B, No.2, pp. 89-124, 1982.

T. L. Magnanti, and R. T. Wong, "Network Design and Transportation Planning: Models and Algorithms," *Transportation Science*, 1984.

H. S. Mahmassani, C. M. Walton, K. C. Mouskos, J. J. Massimi, and I. Levinton, "A Methodology For the Assessment of Truck Lane Needs in the Texas Highway Network," CTR Report 356-3F, Center For Transportation Research, the University of Texas at Austin, 1985.

H. S. Mahmassani, K. C. Mouskos, and C. M. Walton, "Application and Testing of the Diagonalization Algorithm for the Evaluation of Truck-Related Highway Improvements," Presentation at the *66th Annual Meeting of the Transportation Research Board*, January 1987.

H. S. Mahmassani, S. Peeta, G. Chang, and T. Junchaya, "A Review of Dynamic Assignment and Traffic Simulation Models for ATIS/ATMS Applications," CTR Report DTFH61-90-R-00074-1, Center For Transportation Research, the University of Texas at Austin, April 1992.

H. Matsuo, C. J. Suh and R. S. Sullivan, "A Controlled Search Simulated Annealing Method for the Single Machine Weighted Tardiness Problem," Working paper 87-12-2, Department of Management, University of Texas at Austin, 1987.

H. Matsuo, C. J. Suh and R. S. Sullivan, "A Controlled Search Simulated Annealing Method for the General Job shop Scheduling Problem," Working paper 03-44-88, Department of Management, University of Texas at Austin, 1988.

J. J. Massimi, H. S. Mahmassani, and C. M. Walton, "Truck Lane Needs Methodology: A Heuristic Approach to Solve a Five-Option Network Design Problem," CTR Report 356-1, Center For Transportation Research, the University of Texas at Austin ,1985.

Z. Michalewicz, *Genetic algorithms + Data Structures = Evolution Programs*. Addison Wesley, Reading, Massachusetts, 1989.

D. Middleton and K Fitzpatrick, "Truck Accident Countermeasures for Urban Freeways," *ITE Journal*, November 1996.

K. C. Mouskos, "A Tabu-based Heuristic Search Strategy to Solve a Discrete Transportation Equilibrium Network Design Problem," Ph.D. Thesis, Department of Civil Engineering, The University of Texas at Austin, 1991.

K. C. Mouskos, H. S. Mahmassani, and C. M. Walton, "Network Assignment Methods for the Analysis of Truck-Related Highway Improvements," CTR Report 356-2, Center For Transportation Research, the University of Texas at Austin, 1985.

K. C. Mouskos and Q. Zeng, "A Simulated Annealing Heuristic Search Strategy for the User Equilibrium Transportation Network Design Problem," Present at *INFORMS Conference*, Washington, D. C., Spring 1996.

A. Nagureney, *Network Economics*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1993.

G. L. Newhauser, "The Age of Optimization: Solving Large Scale Real-World Problems," *Operations Research*, Volume 42, No.1, January-February 1994.

N. Oppenheim, *Urban Travel Demand Modeling*. Wiley - Interscience, John Wiley & Sons, Inc., New York, New York, 1995.

A. D. Pearman, "The Structure of the Solution Set to Network Optimization Problems," *Transportation Research*, Volume 13B, Number 1, pp. 81-90, March 1979.

H. Pirkul, J. Current and V. Nagarajan, "The Hierarchical Network Design Problem: A New Formulation and solution Procedures," *Transportation Science*, Volume 25, No.3, August 1991.

H. Poorzahedy and M. A. Turnquist, "Approximate Algorithm for the Discrete Network Design Problem," *Transportation Research Board*, Volume 16B, No.1, pp. 45-55, 1982.

C. R. Reeves, "A Genetic Algorithm For Flowshop Sequencing," *Computer Operations Research*, Volume 22, No.1, pp. 5-13, 1995.

W. Rothengatter, "Application of Optimal Subset Selection to Problems of Design and Scheduling in Urban Transportation Networks," *Transportation Research Board*, Vol. 13B, pp. 49-63, 1979.

Y. Sheffi, *Urban Transportation Networks*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1985.

N. F. Stewart, "Equilibrium VS System-Optimal Flow: Some Examples," *Transportation Research*, Volume 14A, pp. 81-84, 1980.