MULTI-OBJECTIVE RAPID TRANSIT NETWORK DESIGN

(ÇOK AMAÇLI HIZLI ULAŞIM AĞI TASARIMI)

by

Erdem GÜNDOĞDU, B.S.

Thesis

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

Date of Submission : May 13, 2010

Date of Defense Examination : June 9, 2010

Supervisor : Assoc. Prof. Dr. Orhan FEYZİOĞLU

Committee Members : Prof. Dr. Tülin AKTİN

Assoc. Prof. Dr. Esra ALBAYRAK

ACKNOWLEDGEMENTS

I am deeply grateful to my supervisor, Dr. Feyzioğlu, for helping and guiding me in completing my thesis. During my study, he did not only lead me about our research, but also encouraged me to struggle with uncertainties. After these two years, I feel that my academic and intellectual knowledge have developed and my approach to different problems has improved significantly. I will be looking for the chance of working with him again in the future.

I would like to thank to my instructors whose courses I had the chance of attending during my master education. These courses are quite useful in the decision of how to approach the problems that I encountered during my research and provide a plenty of tools to solve them.

I also appreciate the instructors in İstanbul Kültür University Industrial Engineering Department that help me improve my academic skills and provide me the necessary conditions to complete this work.

Finally, in the mother's day in which I am writing these sentences, I devote this work to my mother. I am sure this will be the most meaningful gift I can ever give her because she always desires the best for me. I am also grateful to my father for taking care of my problems and not letting me down. This is your success more than mine.

09/05/2010
Erdem GÜNDOĞDU

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Urban rapid transit network design consists of the location of train alignments and stations in an urban traffic context. There are so many criteria that may be taken into account by many decision makers that are involved in design and construction of these alignments because transit networks, like other types of transportation networks, have social and economical affects on the inhabitants of any city or region.

Since the importance of sustainability in urban planning increases in the last decades, authorities pay attention to create more environment-friendly solutions. Air pollution in many cities in the world, especially in highly populated regions, is attributed to the emissions of vehicles. An important mean to reduce these side effects is to enhance public transportation. This study focuses on the design of rapid transit networks because public surface transportation with or without privileged lanes also release harmful gases most of the time. However, it should be noted that the most significant aspect of this type of networks is their required investment costs which are relatively higher than any other means. Moreover, the higher the number of stations involved in the alignment, the more costly the construction will be. In this study, the design of a single transit line with no predetermined origin and destination is investigated while considering the existing road network. The problem is formulated as a bi-level multi-objective optimization problem where the minimization of line investment and vehicle emissions are considered as two separate objectives at the upper level, and user traffic behavior is considered at the lower level. The model reflects the mode choice of the users which are assumed to act rationally by selecting the shortest time path to their destination. The well known multi-objective genetic algorithm NSGA-II is adapted so as to find non-dominated solutions of the problem, and a variant of Frank-Wolfe algorithm is used to solve the lower level traffic assignment problem. The proposed algorithm is applied on a benchmark problem existing in the literature and important insights are provided.

RÉSUMÉ

La conception du réseau urbain rapide des transports détermine l'emplacement où devront se poser les rails des trains et les stations selon les conditions de la circulation urbaine. Comme les réseaux de transport rapide semblables aux autres types de transport, ont des répercussions économiques et sociales sur la population urbaine, il y a plusieurs critères qui doivent être pris en considération par plusieurs personnes qui ont été incluses dans les différentes étapes des travaux de conception et de construction.

Compte tenu de l'importance de la durabilité dans l'accroissement de la planification urbaine durant les dernières décennies, les pouvoirs locaux ont attribué beaucoup plus d'importance pour trouver des solutions qui ne nuisent pas à l'environnement. La pollution de l'air dans plusieurs villes, spécialement dans les régions à forte population, est due à l'émission des véhicules. Un moyen important de réduire ces effets secondaires est d'améliorer les transports publics. Cette étude se concentre sur la conception de réseaux urbains rapides de transport car, la plus part du temps, la surface des transports publics avec ou sans couloirs privilégiés, dégage aussi des gaz nocifs. Toutefois il est juste de noter que l'aspect le plus significatif de ce type de réseaux est dans le coût des investissements qu'ils nécessitent et qui sont relativement plus hauts que ceux de tous les autres moyens. En outre, plus le nombre de stations incluses dans l'alignement sera élevé plus la construction sera coûteuse. Dans cette étude, la conception d'une seule ligne de transport sans origine et sans destination prédéterminées est examinée en même temps que l'on considère le réseau des routes existantes. Le problème est formulé comme un problème d'optimisation à plusieurs objectifs, où la minimisation de ligne d'investissement et les émissions des véhicules sont considérés en tant que deux objectifs séparés au plus haut niveau et le comportement de l'utilisateur de la circulation est considéré au niveau le plus bas. Le modèle reflète le choix des utilisateurs qui sont supposés á être réaliste en choisissant le chemin le plus court pour aller á leur destination. Le fameux algorithme NSGA-II à objectifs multiples est adapté de façon à trouver des solutions non dominantes au problème et une variante de l'algorithme Frank-Wolfe a été utilisée pour résoudre le problème de l'affectation de la circulation de bas niveau. L'algorithme proposé est appliqué sur un problème de critères qui existent dans la littérature et d'importantes argumentations sont fournies.

ÖZET

Kent içi hızlı ulaşım ağı tasarımı, mevcut kentsel trafik şartlarında, ray hatlarının ve istasyonların konumlandırılacağı yerleri belirler. Diğer ulaşım tiplerine benzer bir şekilde hızlı ulaşım ağlarının, kent halkı üzerinde sosyal ve ekonomik etkileri bulunduğundan, tasarım ve yapım aşamalarında konuya dâhil olan birçok kişi tarafından dikkate alınması gereken birçok ölçüt vardır.

Son yıllarda kentsel planlamada sürdürülebilirliğin önemi arttığından, uzmanlar daha çevre dostu çözümler bulmaya dikkat etmektedirler. Başta yoğun nüfuslu bölgeler olmak üzere birçok şehirde hava kirliliği araçlardan çıkan gazlarla ilişkilendirilir. Bu yan etkileri azaltabilmenin önemli bir yolu toplu taşımayı teşvik etmektir. Otobüs ve ya metrobüs gibi toplu taşıma araçları da zehirli gazlar saldığından dolayı, bu çalışma hızlı ulaşım ağları üzerine yoğunlaşmaktadır. Ancak bu tip ağların en belirgin özelliğinin, diğer toplu taşıma tiplerine oranla daha yüksek bir yatırım maliyetine sahip olması olduğunu belirtmek gerekir. Dahası, istasyon sayısı arttıkça, yapım maliyeti de artacaktır. Bu çalışmada, mevcut araç ağını da dikkate alarak, başlangıç ve bitiş noktaları önceden belirlenmemiş tek bir hattın tasarımı incelenmiştir. Problem, gaz salınımının ve yapım maliyetinin iki ayrı amaç olarak üst seviyede en azlanacağı, kullanıcıların trafik davranış eğilimlerinin ise alt seviyede incelendiği çok ölçütlü iki seviyeli bir model olarak tasarlanmıştır. Model, gidecekleri noktaya mümkün olan en kısa yolu seçerek gerçekçi davranacağı varsayılan kullanıcıların seçimlerini yansıtmaktadır. Baskın çözümleri bulabilmek amacıyla, NSGA-II çok ölçütlü evrimsel algoritması probleme uyarlanmış ve alt seviyedeki trafik atama problemini çözebilmek için de Frank-Wolfe Algoritmasının bir uyarlaması kullanılmıştır. Önerilen algoritma, literatürde çokça kullanılan bir probleme uygulanmış ve önemli çıkarımlar sunulmuştur.

1 INTRODUCTION

Public transportation is one of the main services that inhabitants of a city or a region require for not only traveling but also other economic and social issues. This is because, public transportation systems, in fact well-designed public transportation systems may help decrease air pollution, reduce traffic congestion and as a result travel time, besides providing a cheaper way of traveling. Of course, the design of these systems and methods to provide it, are of great importance. For instance, if these systems were conducted by only buses using the roads of private vehicles, it would provide no benefit to use public transportation because of the congestion when the maximum capacity of links is reached. However, if rapid transit networks are available, users will be encouraged for using public transportation because of its short travel time and low cost comparing to the traveling by private vehicles.

For such reasons, in cities especially with high levels of population, the portion of budgets dedicated to rapid transit network investments increase in the last decades. This type of networks is also preferred because of that it reduces air pollution much more than the classical transportation tools such as buses or buses with privileged lanes.

However, it is very costly to construct such networks even with few stations and municipalities may not be eager to spend huge amounts on this type of investment. The problem here is determining an optimum solution between sustainability issues, that is, the amount of emission decreased by constructing this network and available budget of municipality while taking user behavior into account. In this study, it is aimed to find a solution to the problem described above. Our study differs from other ones in such aspects;

- To the best of our knowledge, this is the first study that takes environmental issues into account in constructing a rapid transit line. Since air pollution is a serious problem and traffic congestion is an important source of this pollution, it is aimed to construct a line that minimizes the emission of the vehicles as much as possible. Because the number of vehicles, traffic congestion and gas emission are not at the same level in each link, the route which the line will pass through, nodes that stations will be located at and decision of which OD pairs will be connected by a transit line must be planned carefully.

- Travel times of the links are not constant in this study. That means, for every possible solution (a solution is a rapid transit line with stations located.) travel times are computed considering the users' choice because according to the travel times of transit line and private road network, users choose their travel modes among these alternatives by comparing their travel times.

- Most of the studies until now use one objective such as maximizing trip coverage and population covered or a normalized equation including these objectives with predetermined weights. Since the problem is a tradeoff between minimizing emission and minimizing cost and the effectiveness of the solution is determined by user behavior, a multi-objective bi-level model is proposed in this study. This problem is actually a Stackelberg game in which the municipality is the leader that makes decisions and the followers are the users that can make their choices when a rapid transit line is constructed according to the priorities of the municipality. The upper level problem (leader) has two objectives mentioned above and the lower problem (followers) aims to find user equilibrium in the entire network. That means users will change their transportation modes until all alternatives have the same travel time between the same OD pairs.

The organization of this study is as follows; Chapter 2 reviews previous studies on RTNDP dealing with both station locations and generating alignments. In Chapter 3, methods and algorithms used are elaborated and at the end of this chapter our multi-objective approach to RTNDP is presented. Chapter 4 is dedicated to the case study of our algorithm which is a well-known problem, Sioux-Falls Networks. Mathematical model is provided in Chapter 5 and our objective functions and constraints are explained in detail and finally, conclusion and future directions are described in Chapter 6.

2 LITERATURE REVIEW

Transit Network Design Problem (TNDP) deals with the optimal design of the routes and frequencies in urban public transit systems. It has two main sub-problems; the first one is continuous of which subject is expansion of existing links. The other one, discrete transit network design, deals with adding new links to the original network. Rapid Transit Network Design Problem (RTNDP) is included in the latter one and it finds the optimal line (metro, railway etc) and locations of stations on it.

2.1 NETWORKS, LINK PERFORMANCE FUNCTIONS AND USER EQUILIBRIUM

The term network is commonly used to describe a structure that can be either physical (e.g. streets and intersections, telephone lines and exchanges, etc.) or conceptual (e.g. information lines and people, affiliation relationships and television stations, etc.). Each of these networks includes two types of elements: a set of points and a set of line segments connecting these points. This observation loads to the mathematical definition of a network as a set of nodes (vertices, points) and a set of links (arcs, edges) connecting these nodes.

Each network link is typically associated with some impedance that affects the flow using it. The units of impedance depend on the nature of the network and the link flows. Impedance can represent electrical resistance, time, cost, utility and other relevant measure.

The networks discussed in RTND problem are typically connected. In other words, it is possible to get from any node to any other node by following a path (route) through the network. A path is a sequence of directed links leading from one node to another.

The movement of vehicular traffic through streets and intersections is not the only flow in the urban area. A transit link can be represented by a simple linear network in which the transit stations are represented by nodes and the line haul-portion by links. The impedance on each of these links includes "in-vehicle" travel disutility elements such as travel time.

The transportation planning process for urban areas is typically based on a partition of the area into traffic zones. The size of each zone can vary from a city block to a whole neighborhood or a town within an urban area. The number of traffic zones can vary from several dozens to several thousands. Each traffic zone is represented by a node known as centroid. The centroids are those source and sink nodes where traffic originates and to which traffic is destined. Once the set of centroids is defined, the desired movement over an urban network can be expressed in terms of an origin-destination (O-D) matrix. This matrix specifies the flow between every origin centroid and every destination centroid in the network.

The travel impedance or level of service, associated by links representing an urban network can include many components, reflecting travel time, safety, cost of travel, stability of flow and so on. The primary component of this impedance is, however, is travel time which is often used as the sole measure of link impedance. Generalized impedance, which combines several measure can be used and the term travel time can be understood as such a combined impedance.

The level of service offered by many transportation systems is a function of the usage of these systems. By congestion, travel time on urban streets and intersections is an increasing function of flow. Consequently, a performance function rather than a constant travel-time measure should be associated with each of the links representing the urban network. The performance function relates the travel time on each link to the flow traversing this link.

The travel time with zero flow is known as the free-flow time. At this point, a traveling car would not be delayed because of interaction with any other car moving along the link. The only source of delay at this point is the time associated with traversing the link (and the probability of being stopped by a red signal indication). As the flow increases, the travel time monotonically increases since both the travel time along the approach increases and the intersection delay increase with the flow. Characteristically, the performance function is asymptotic to a certain level of flow known as the capacity of the transportation facility under consideration. The capacity is the maximum flow that can go through any transportation capacity. The performance function is undefined for higher values of flow, since such flows can not be observed.

The general shape of the performance functions is similar for links representing most types of urban streets. The physical characteristics of each street such as length, width, parking restrictions, turning pockets and signal green time, determine the exact parameters of the function for each street.

In this study, performance function for each link is identical. A simplified function that is often used in practice is the equation developed by the U.S. Bureau of Public Roads (BPR). This equation is given by:

$$t_{ij} = t_{ij}^0 \left[1 + \alpha \left(\frac{x_{ij}}{c_{ij}} \right)^{\beta} \right] \tag{2.1}$$

In this formula, t_{ij} and x_{ij} are the travel time and flow respectively on link a. t_{ij}^0 is the free-flow time and c_{ij} is the capacity of link a. The quantities α and β are model parameters, for which the value $\alpha = 0.15$ and $\beta = 4$ are typically used. These values imply that the practical capacity of a link is the flow at which the travel time is 15% higher than the free-flow time. This does not equal the capacity of the road, that is, the maximum possible flow through a link.

The only travel choice in a networks consisting of just a road network is the motorists' choice of routed between their origins and destinations. The problem can be put as follows:

Given:

1. A graph representation of urban transportation network
2. The associated link performance functions
3. An origin-destination matrix

This problem is known as traffic assignment since the issue is how to assign the O-D matrix to the network. To solve the traffic assignment problem, it is required a rule by which the motorists choose a route be specified. The interaction between the routes chosen between all O-D pairs, on the one hand, and the performance functions on all the network is on the other determines the equilibrium flows and the corresponding travel times throughout the network.

It is reasonable to assume that every motorist will try to minimize his or her own travel time when traveling from origin to destination. However, this does not mean all travelers between each origin and destination pair should be assigned to a single path. The travel time on each link changes with the flow and therefore, the travel time on several of the network path changed as the link flows change. A stable condition is reached only when no traveler can improve his travel time by unilaterally changing the routes. This is the characterization of the user-equilibrium (UE) condition.

The user equilibrium condition assumes that motorists have full information that means they know the travel time on every possible route and that they consistently make the correct decisions regarding route choice. Furthermore, it assumes that all individuals are identical in their behavior. These presumptions can be relaxed by making a distinction between the travel time that individuals perceive and the actual time. The perceived travel time can be looked upon as a random variable distributed across the population of drivers. In other words, each motorist may perceive a different travel time, over the same link. Equilibrium will be reached when no traveler believes that his travel time can be improved by unilaterally changing routes. This definition characterizes the stochastic-user-equilibrium (SUE) condition [1].

2.2 NETWORK DESIGN

Network design deals with designing the transportation network. Public transportation networks (i.e. bus routes, train lines) are determined by using network design tools. In most of the real life cases, existing public transit networks are modified. These modifications can include;

- finding new stations in a railway or bus network,
- closing existing stations,
- finding a sub-network for opening rapid transit lines [2]

The Network Design Problem (NDP) can be classified in according to four criteria: (1) system-optimal (SO) / user optimal (UO) behavior (2) static/dynamic traffic assignment (3) discrete/ continuous investment variable (4) deterministic/stochastic parameters. The SO behavior is mathematically tractable but unrealistic. The static traffic assignment assumes the steady-state condition, whereas the dynamic traffic assignment (DTA) accounts for time dynamics. The discrete investment variable allows entire-lane or new link addition. The continuous investment permits a fraction of lane addition. The continuous investment variable has extensively been employed in the literature and the justification is that because most roads in the urban area are already constructed. The continuous link expansion can be implemented by altering lane width, median and shoulder area. Alternatively, the continuous NDP can be considered as a possible heuristic for the discrete NDP [3]. Lastly, the problem parameters have typically considered deterministic where all users have perfect information about the travel times of each alternative, as opposed to stochastic where the travel time of the same link may be perceived differently by different users.

The flow pattern that minimizes SO program does not generally represent an equilibrium condition. Except in some special cases, it can result only from joint decisions by all motorists to act so as to minimize the total system travel time rather than their own. In other words, at the SO flow pattern; drivers may be able to decrease their travel time by unilaterally changing routes. Such a situation is likely to sustain itself and consequently SO flow pattern is not stable and should not be used as a model of actual behavior and equilibrium.

The significance of SO formulation and the resulting flow pattern is that the value of the SO objective function may serve as a yardstick by which different flow patterns can be measured. Indeed system wide travel time is a common measure of performance of a network under a given scenario. This measure can be computed in a straightforward manner given the equilibrium flows and it does not require any data in addition to those required for the equilibrium analysis itself. Thus the flow pattern associated with any proposed project can be measured in terms of the total travel time associated with it relative to the minimum possible travel time. This program can be expressed as follows:

$$\min \tilde{z}(x) = \sum_{(i,j)\in A} x_{ij} t_{ij}(x_{ij})$$

$$\text{subject to}$$

$$\sum_k f_k^{rs} = q_{rs} \qquad \forall r,s$$

$$f_k^{rs} \geq 0 \qquad \forall k,r,s \tag{2.2}$$

However, it is reasonable to assume that every motorist will try to minimize his or her own travel time when traveling from origin to destination. The travel time changes in each link changes with the flow and therefore the travel time on several of the networks paths change as the link flows change. A stable condition is reached only when no traveler can improve his travel time by unilaterally changing routes. In other words, for each OD pair, the travel time on all used paths are equal and also less than or equal to the travel time that would be experienced by a single vehicle on any unused paths. This is the characterization of the user equilibrium (user optimum) condition. The objective function of UO is the minimization of;

$$\sum_{(i,j)} \int_0^{x_{ij}} t_{ij}(w)dw \tag{2.3}$$

subject to the same constraints of SO.

In this study, a transit line is to be constructed from scratch in such a way that minimizes the total gas emission releasing from the vehicles in private network. It is assumed that traveling by bus is not an alternative so this study mostly deals with the third item. It is modeled as steady-stated because peak time of the day is used for analysis, deterministic because each user perceive the travel times of each link and alternative the same and discrete because decision to be made is to connect two nodes by transit line or not. The model uses UO which is more realistic comparing to SO in the lower problem where total travel time is minimized.

2.3 RAPID TRANSIT NETWORK DESIGN

The Rapid Transit Network Design problem consists of the location of train alignments and stations, in context where the demand makes its own decisions about the mode and the route [4]. In this context, the core issue-location of a single line and corresponding stations is a classical multi-objective network design problem [5]. For the design of the rapid transit line, both quantifiable and non-quantifiable criteria have to be considered: construction and operating costs, travel times, demand satisfaction, utilization and accessibility of open space and historic sites, air and noise pollution and so forth [6]. Decisions are the consequences of the interaction of several actors: engineers, central and local administrators, environmentalists and lobbies [5]. Because of the very large cost of constructing and operating rapid transit systems, it is important to pay close attention to their efficiency and effectiveness [7]. The Rapid Transit Network Design Problem contains two intertwined problems: determining an alignment and locating stations on it [8].

Bruno et al [6] compares his model with Maximum Covering Shortest Path Problem (MCSPP) and the Median Shortest Path Problem (MSPP) and bi-level programming models are introduced for generation of efficient solutions considering main criteria. The MCSPP deals with the minimization of the total path construction cost and the maximization of the total demand satisfied (sum of the nodal demands covered by this path). The criterion of coverage, which supposes that a nodal demand d_k is satisfied or covered if a RTL station is located within a given fixed distance from node k, is the basis of the formulation of the MCSPP. The MSPP objectives are the minimization of the total construction cost, like MCSPP, and the maximization of the accessibility of the path. However, because of the lack of realistic assumptions about the users' behavior, neither model provides satisfactory results. In this study, a bi-criterion path location model with predetermined origin and destination nodes is proposed. It assumes that each user chooses the path with the least travel cost among different alternatives. Travel cost includes travel time, monetary cost and comfort. In particular, the model is a sort of generalization of MSPP.

There are 4 different networks in this study: private, public, pedestrian, and pedestrian-public. Users can choose any of these travel modes comparing their travel costs. The model is mainly based on these assumptions:

- Mobility demand is described by an origin-destination matrix.
- Users can use private mode or the hybrid pedestrian-public transportation system to be designed.
- Demand is assigned to the transportation system corresponding to the least travel cost.
- Travel costs are independent of flows.
- Mobility demands and cost are deterministic.

To estimate the set of non-inferior solutions, the procedure identifies K shortest paths on the public network with starting node "o" and terminus "d" using a label correcting technique, referring to construction costs, z_1. For each generated path, the corresponding bi-level network is built by augmenting the pedestrian network with the arcs and nodes of the path. Then, the total weighted travel cost incurred by the users, z_2 associated with that path is calculated. Finally, the efficient solutions are selected in the set of the generated paths using the dominance relationships.

Hamacher et al [9] analyzed the effects of adding new train stops to an existing rapid transit line by using genetic algorithm. The problem is a tradeoff between the number of people attracted by adding some extra stops and people that change the travel mode because of longer travel times. It is also shown that the problem is NP-hard.

Let P denote the set of coordinate points, each of them representing a settlement and let L be the set of feasible points along the tracks of the railway company. A solution X to the problem is given by a set of points in the plane, representing the stops that should be installed. X is feasible if $X \subseteq L$, i.e., if each $x \in X$ satisfies $x \in L$. To evaluate a feasible solution X, we are interested in how many settlements are covered by X. A settlement $p \in P$ is covered by a point $x \in L$ if;

$$l_2(p, x) \leq r \tag{2.4}$$

where l_2 refers to the Euclidean distance and r is some given radius. Basically, the number of new stops is interested in, it is aimed to cover as many settlements as with as few stops as possible.

It is supposed that L is given as a network $G = (V, E)$ embedded in the plane. For a point $x \in L$, $e(x) \in E$ denotes the edge of G on which x is located. Furthermore, for each edge e, the number of customers traveling along e is denoted by w_e. Then for a set X of new stops,

$$t_d(X) = \sum_{x \in X} s.w_{e(x)} \tag{2.5}$$

gives the amount of additional travel time for customers, which is caused by the additional stop activities of the trains. It is assumed that a constant time delay s is caused by any additional stop of train.

On the other hand, some of the customers will save travel time, since a new train stop may reduce the distance to their closest train stop and hence the time they need to get on a train. In the model, the reduction of the distance for a point $p \in P$ is calculated by,

$$l_2(p, S) - l_2(p, S \cup X) \tag{2.6}$$

where $S \subseteq V$ denotes the set of already existing train stops and $l_2(p, Y)$ is the closest Euclidean distance from p to any point in Y.

To transform the possible reduction of distance into an amount of saved access time a piecewise linear function is introduced $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ in two variables, assigning an amount of saved time to each reduction of distance, given as a pair old and new distance of a settlement from the nearest train stop.

Denoting for each $p \in P$ the number of customers from the corresponding settlement by v_p, the positive effect to the travel time through saved access time can be calculated by

$$t_a(x) = \sum_{p \in P} v_p . g(l_2(p, s), l_2(p, s \cup X)) \tag{2.7}$$

The travel time model can be summarized:

$$\max t_a(X) - t_d(X)$$

such that

$$x \in L \text{ for all } x \in X \tag{2.8}$$

The change in train riding time that is caused by starting or ending the trip at a different train stop is neglected, assuming that these gains and losses in train riding time roughly even out.

Laporte et al [10] aims to locate stations on a given line so as to maximize the coverage of people by the stations. The coverage provided by a station situated in a central area will depend on the pedestrian network around it and the limits of the coverage will be determined by the maximum distance that users are able or decide to walk the transit network. The percentage of covered users depends on the required time to reach the nearest station, thus several levels of attraction are stated.

The number of stations to be located in a section with fixed extremes is a function of the length and inter-station spaces. There are two parameters, s_{min} and s_{max}, which are bounds on the minimum and maximum inter-station spacing. The number of stations to be located is

$$N = \lceil L/\bar{s} \rceil \tag{2.9}$$

where L is the length of the section and $\bar{s}$ is the average inter-station space. N is fixed and the distance between two adjacent stations is assumed to belong the interval $[s_{min}, s_{max}]$ which is called "constraint on inter-station spacing" (CIS).

C is the catchment area of the transit line, divided into census tracts c_j such that;

$$C = \cup_{j=1}^{j} c_j \tag{2.10}$$

where each zone c_j is assumed to be a polygonal region with population density ρ_j, $\forall j = 1, \ldots, J$. If data about jobs are available, ρ_j could be reinterpreted as a sum of the population and employment density. $d(x, y)$ is a distance measure in the plane with $||.||$ as associated norm $d(x, y) = ||x - y||$. The set of points in the plane whose distance to the station x_i is not greater than r (usually called the ball of radius r) is denoted by $B(x_i, r)$. For each station x_i, K different levels of attraction is considered. Then, $\forall i = 1, \ldots, N$ and $\forall k = 1, \ldots, K$

$$v \in B(x_i, r_k) = B_{ik} \text{ if and only if } ||v - x_i|| \leq r_k. \tag{2.11}$$

These attraction levels correspond to concentric annuli around each location site.

There are several studies about modeling the attraction by function. In this study, the function derived by the gravitational model is used.

$$\theta(z) = \frac{a}{z^2}, \qquad z \in [d_{min}, d_{max}], \qquad 0 < d_{min} < d_{max} \tag{2.12}$$

Here, $\theta(d(x, x_i)) = a/||v - x_i||^2$ in which a is a constant to be calibrated.

In order to discretize the objective function, the attraction in each annulus, determined by two consecutive levels, will be considered constant. Thus, the coverage provided by the station located at x_i is

$$R(x_i) = \sum_{j=1}^{J}\sum_{k=1}^{K} \frac{a\rho_j}{(r_{k-1} + r_k/2)^2} \text{Area}((B_{ik} \backslash B_{i(k-1)}) \cap c_j \tag{2.13}$$

Finally, the objective function is the cover provided by the stations situated in alignment l, which in the plane corresponds to the section of the line under consideration subject to CIS. This yields

$$\max_{x_i X \cap (CIS)} R(l) := \sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{k=1}^{K} \frac{a\rho_j}{(r_{k-1} + r_k/2)^2} \text{Area}((B_{ik} \backslash B_{i(k-1)}) \cap c_j) \tag{2.14}$$

In the study of Bruno et al. [11], a two-phase heuristic is proposed and objective is defined as the maximization of the total population covered by the alignment. This objective is used because obtaining reliable O/D demands is costly and travel demand available at the network planning stage may differ from the actual demand observed after the system has been built. However, population coverage is an approximation because the use of the public transportation also depends on the factors such as income level, car ownership, etc. It is assumed that all population associated with a vertex where a station is located uses the transit system. A grid network is used and it is assumed that a population $p(x, y)$ is associated with each integer coordinate vertex (x, y). The population covered by a station s, called the cover of s, is defined as

$$C(s) = \sum_{d=0}^{3} \sum_{(x,y):D[(x,y),s]=d} = \theta_d p(x, y) \tag{2.15}$$

where $D[(x,y),s]$ denotes the Manhattan distance between (x,y) and s.

Distances between consecutive stations are also restricted by a constraint in the model. l_{min} and l_{max} are the minimum and maximum distances allowed respectively. This constraint ensures a balance between the additional time by stopping in a station and additional number of users attracted by the located station.

The heuristic proposed consists of two phases; construction of an alignment and solution improvement. The first phase constructs a feasible alignment by adding one station at a time. It gradually extends a partial alignment $X_t = (s_1, \ldots, s_t)$ by first determining the location s_p outside X_t having the largest cover $C(s_p)$ and then checking whether s_p is $(n-t-1)$-linkable with the nearest of s_1, that means s_p and s_1 can be linked by a partial alignment containing at most k intermediate stations satisfying the interspacing constraints. In the second phase, an attempt is made to improve upon the best known alignment X^* by extracting from it a partial alignment and extending it into several full alignments.

Laporte et al [8] estimates actual demand between two nodes and then, generates an alignment that maximizes the trip coverage (total station-to-station ridership covered by the alignment) by using greedy heuristics proposed. An alignment location methodology based on O/D information is presented. O/D information is said to be needed because other methods such as corridor approach and catchment areas are unsatisfactory. Corridor approach assumes that someone living near close to metro line but far from a station will be attracted by this line which is obviously not the case. Catchment area method also has a major drawback. According to this method, someone travelling along North-South axis will be attracted by the station located in an East-West alignment. Planners use the data provided by the census tracts to evaluate the population covered by a station. Let $C = \{c_h : h = 1, \ldots, H\}$ be the set of census tracts and ρ_h the respective density of each c_h; $h = 1, \ldots, H$. Then the coverage $R(i)$ provided by the station s_i is defined by;

$$R(i) = \sum_{h=1}^{H} \sum_{k=1}^{K} \frac{\alpha \rho_k}{\bar{r_k}^2} \text{Area}((B_k \backslash B_{k-1}) \cap c_h); \quad i = 1, \ldots, n, \tag{2.16}$$

where k is a ring index, K is the number of rings per station, $B_k \backslash B_{(k-1)} \equiv B(s_i, r_k) \backslash B(s_i, r_{k-1})$ is the ring centered at s_i and comprised between radii r_{k-1} and r_k, $\bar{r_k}$ is an intermediate value between radii r_{k-1} and r_k.

To combine the estimation of passenger O\D patterns with the notion of population coverage, Mesa et al [10] proposed the following approach.

Each station pair (i, j) has an associated $K \times K$ matrix, denoted by OD_{ij}, whose elements $OD_{i,j}(k, k')$ (for all $k, k' = 1, \ldots, K$) represent the weighted sum of portions of values t_{lm} (predicted the number trips produced in zone z_l and attracted to zone z_m, for all $l, m = 1, \ldots, L$). Weights are defined by taking into consideration attraction radii r_i and $r_{i'}$ (respectively, at origin and destination stations) yielding the following O\D values:

$$OD_{ij}(k, k') = \sum_{l,m=1,(l \neq m)}^{L} \frac{a^2}{\bar{r_k}^2 \bar{r'_k}^2} \frac{\text{Area}((B_{jk'} \backslash B_{j(k'-1)}) \cap z_m}{\text{Area}(z_m)} t_{lm} \tag{2.17}$$

Therefore, the trip coverage f_{ij} provided by a pair of different stations (s_i, s_j) can be obtained by adding all matrix elements:

$$f_{ij} = \sum_{k,k'} OD_{ij}(k, k'). \tag{2.18}$$

The model assumes that potential users choose between only two transport modes; a transit line $Align(E)$, defined on a set of E stations and private car that uses the street grid. The proportion of trips between s_i and s_j, using the rapid transit network on set E is obtained by;

$$g_{ij}(E) = \frac{1}{1 + e^{-\gamma(\tau_{ij}^A - \tau_{ij})}}(E) \tag{2.19}$$

where $\tau_{ij}(E)$ denotes the time of the quickest path connecting s_i and s_j using the network based on E, τ_{ij}^A represents the time of the quickest path connecting s_i and s_j by using another means of transport and γ is a positive parameter to be calibrated.

The objective of the model can be described as follows:

$$(MCTP) = \begin{cases} \max\limits_{E \subset S} \sum\limits_{i,j(i \neq j; s_i, s_j \in E)} n_{ij}(E) \\ subject\ to \quad Length[Align(E)] \leq LMAX \end{cases} \tag{2.20}$$

$n_{ij}(E)$ can be used to represent the passenger flow between s_i and s_j in the presence of an alignment on E. Hamiltonian path is used for generating the alignment and alignment is forced to be at most 20 km.

The first heuristic may be viewed as a greedy algorithm.

1. Include in $Align(E)$ the edge (s_i, s_j).
2. Extend the current alignment $Align(E)$ at either end by adding a new edge, (s_k, s_i) or (s_j, s_k), yielding the maximum objective $\sum\limits_{\forall s_i, s_j \in E} n_{ij}(E)$, not intersecting the current alignment and not causing $Length[Align(E)]$ to exceed $LMAX$. Repeat until the current alignment can no longer be extended.

The second heuristic is the extension of an alignment through greedy stations. That is;

1. Include in $Align(E)$ the edge (s_i, s_j) yielding max n_{ij}.
2. Consider in turn for each non-inserted station and insert it in the current alignment.

After applying the second heuristic, solution may be improved by a post-optimization procedure.

1. Given a solution, remove in turn each station, link its predecessor to its successor and reinsert the station just removed.
2. Stop when no station removal and reinsertion can yield an improvement in objective function.

These heuristics are also applied on real data, Sevilla metro and the results are compared.

Laporte et al [13] describes three stages of the process of designing a rapid transit system as selecting key nodes, designing the core network and locating secondary stations. It is aimed to maximize coverage and designing the core network is achieved by using a short list of lines, which are supported by the key nodes previously selected, that means second stage is addressed. Key nodes are selected by using weighted demands of nodes according to their number of inhabitants and visitors among candidate locations such as hospitals, universities, commercial zones, etc.

After having located the stations that must belong to some of the lines of the network, the problem of connecting them with a small number of alignments $A = \{A_l : \ l = 1, \ldots, L\}$, with origins o_l and destinations d_l given, in competition with the private mode, is tackled.

Let E be the set of feasible edges linking the key stations. Therefore, there is a network $N = (N, E)$ from which the core network is to be selected. For each node $n_i \in N$, the set of nodes adjacent to node i is denoted by $N(i)$. Let c_{ij} and c_i denote the costs of constructing a section of an alignment on edge ij and that of constructing a station at node n_i. The generalized routing cost (under demand point of view) of satisfying the demand of pair p through the private and public network are c_p^{PRIV} and c_p^{PUB}, respectively. The first one is a given value, but the latter cost depends on the final topology of the public network and therefore on the edges that are selected; for this reason a generalized cost c_{ij}^{PUB} is given for each edge. This value is taken equal to the distance between i and j. Depending on the available budget for the total construction cost of each alignment, bounds c_{min}^{l} and c_{max}^{l}, $l = 1, \ldots, L$ on the construction cost of each alignment and bounds c_{min} and c_{max} on the total construction network cost are known.

The problem consists of choosing a low number L of lines $(1 \leq L \leq 5)$ covering as much as possible the travel demand between the points of N, subject to constraints on the construction cost.

In the formulation of the model, there is not a constraint for sub-tour elimination, but when a solution contains a cycle, then such a constraint is imposed. The model takes cost into consideration via a constraint instead of an objective function.

Marin et al [4] extends the study of Laporte et al [8] and incorporates the line locations constraints with a bounded but variable number of lines with no predetermined origins and destinations. The model proposed, Extended Rapid Transit Network Design (ERTND), has a greater degree of freedom: the number of lines and their origin destination are variables within the bounds. The lines are not initially given and they do not have fixed origin and destinations.

Key stations are selected in a similar manner to [8]. The set E of feasible edges linking the key stations N is defined. That means, there an undirected graph $G(N, E)$ from which the rapid transit network is to be selected.

The set of possible links is a subset defined by $\{(i, j) : i < j, i, j \in N\}$ of the set of all bidirectional links. That is, (i, j) and (j, i) are identical because the links are assumed to be undirected edges. The demand is, again, given by O\D pairs of nodes as in [8]. The matrix $d = d_{ij}$ of distances between pairs of nodes is used to define public cost and since the system is designed to be underground, the values of d matrix could correspond to Euclidean distances. c_{ij} and c_i are the costs of constructing an edge (i, j) and station at node i, respectively. There are cost upper and lower limits for each line,

c^l_{min} and c^l_{max} and for the entire network c_{min} and c_{max}. Finally, the users make their decisions considering the generalized cost of satisfying it through the public and private networks which are u^{pri}_w and u^{pub}_w. The first one is an input data while the latter one depends on the final topology of the network.

The objective functions of the model are;

- Maximize the public trip covering, z_{pub}.
- Minimize routing cost upper bound, $\bar{z}_r$.

The overall objective function is;

$$z = -\eta z_{pub} + (1 - \eta)\bar{z}_r \tag{2.21}$$

where η is typically a number close to 1. As it can be seen, the overall objective function is a normalization of two objectives rather than a multi-objective programming.

Up to this point, there is not much difference than the model of [8]. However, the problem is extended by adding new constraints which is named as ERTND.

The extension of RTND studies the following topics;

- The number of lines is variable within a given bound.
- The lines do not have predetermined origins and destinations, all nodes in the network are available.

The design of Marin et al [14] incorporates with the fact that users can choose their transportation modes and trips. Objective functions consist of maximizing the public transportation, minimizing routing cost and minimizing locating costs, z_{pub}, $\bar{z}_r$ and z_{loc} respectively. These objective functions are then normalized in order to be integrated in a single objective function as follows:

$$z = -\eta z_{pub} + \frac{1-\eta}{4}\bar{z}_r + \frac{3(1-\eta)}{4} z_{loc} \tag{2.22}$$

Several variations of Benders decompositions are implemented in order to accelerate the convergence towards a solution. The problem has two decision levels: first, the system operator chooses network location and the second; the users choose mode and routes. To apply Benders decomposition, the problem is divided into a master problem defining a feasible network (the location problem) and a sub-problem assigning demand to this network. Benders decomposition iterates between the Master Model and the Sub Model to find an optimal solution. In each iteration k, the dual variables of the SM define optimality or feasibility Benders Cut, which are added to the constraints of the master problem. The process continues until it converges under convexity assumptions verified by the model.

Aggregated Benders Decomposition is the classic Benders algorithm and the base for the next convergence-acceleration extensions. The Benders convergence assumptions require that the SM be a convex problem and then it is necessary to relax the sub model variables.

Extended rapid transit network design problem is modified in [15] to allow circular lines. A two-stage approach is proposed and compared to the results obtained from the model of [4].

Let $L = \{1, \ldots, 1\}$ be the set of potential lines to be constructed. Theoretically, q should be large enough to allow the achievement of maximum possible total expected number of users. If there are λ lines going through a location i or linking two stations i and j, then the associated construction costs are λa_i and λc_{ij} corresponding to constructing a station and linking locations respectively.

The model considered in [4] is modified since that model allows cycle consisting of more than one line (for example, two lines of which endpoints coincide), in this case, it would be preferable to define a unique circular line as the union of the initial lines, since this would reduce the construction costs at their endpoints, as well as the number of transfers that should be done by the users to arrive their destinations.

In the first stage, an integer model is solved for selecting the stations and the links between them are determined and in the second stage the line design problem is solved by means of a procedure that assigns each selected link to exactly one line under certain constraints.

There are two models proposed in this study with the same objective function used in [4]. An undesirable property of the feasible solutions for the first model is that given two opposite pairs $w = (o_w, d_w)$ and $\bar{w} = (d_w, o_w)$, the paths recommended to the users of these pairs have distinct total lengths. Another undesired property is that the users of an O\D pair of locations can be recommended to follow a path that goes through the same location more than once (i.e. contains a circuit). It is assumed that the users of pair w will utilize the rapid transit network if and only if total distance covered by the users of pair w whenever they follow the recommended path is less than or equal to generalized cost of satisfying the demand through an existing network.

Model 2 is a modified version of Model 1. Model 1 allows the possibility of more than one linking two locations. However, if there were λ lines linking two locations, where $\lambda \geq 2$, it can easily be shown that it would be possible to eliminate all but one of those links and redefine the lines for the rapid transit network in such a way that its number would be increased be $\lambda - 1$ units at most and obviously, this would reduce the total construction cost without modifying the value of the objective function for the considered feasible solution of Model 1. Therefore any two locations can be connected by at most one line.

In our model, the upper problem contains minimizing the construction cost and minimizing the gas emission raised from the vehicles on private network, while the user traffic behavior is considered at the lower problem. Since construction of a rapid transit line is quite expensive, it should be designed effectively.

Table 2.1 A Review of the Studies in RTNDP

	Bruno(98)	Hamacher(01)	Bruno(02)	Laporte(02)	Laporte(05)	Laporte(07)	Marin(07)	Marin(09)	Escudero(09)	Our Model(10)
alternative modes	yes	yes	no	no	no	yes	yes	yes	yes	yes
Multiple\single objective	multiple (bi-level)	single	single	single	single	single	two objectives, normalized in a single equation	three objectives, normalized in a single equation	single	multiple
Multiple\single lines	single	single	single	single	single	many	many	many	many	single
circular lines allowed	no	no	no	no	no	no	no	no	yes	no
objective function(s)	minimizing construction cost, minimizing total user travel cost	maximize travel time difference between new network and existing network	maximize population coverage	maximize weighted coverage	maximize trip coverage	maximize trip coverage	maximize trip coverage, minimize routing cost upper bound	maximize trip coverage, minimize routing cost upper bound, minimize location cost	maximize trip coverage	minimize emission minimize cost
OD pairs predetermined	yes	extension of an existing line	no	locating stations on a predetermined line	no	no	no	no	no	no

According to Laporte et al [13] the classical 4-stage model for forecasting the travel demand is as follows;

1. Trip Generation Analysis: computation of the number of trips starting in each zone for each particular trip purpose.
2. Trip Distribution Analysis: production of a table containing the number of trips starting in each zone and ending in each other zone.
3. Mode Choice Analysis: allocation of trips among the currently available transportation systems.
4. Trip Assignment Analysis: assignment of trip flows for the specific routes on each transportation system that will be selected by the users.

This study primarily addresses the last two stages. There are two alternatives for the users; private network which is the road network for the vehicles and the rapid transit line. Users will select their routes and modes according to the travel times in each mode. It is aimed to design a rapid transit network that minimizes the gas emission and construction cost of the line.

2.4 VEHICLE GAS EMISSION

Gas emission released from the vehicles is an important source of air pollution. The most important pollutants emitted by road vehicles include ozone precursors, greenhouses gases, acidifying substances, particulate matter mass, carcinogenic species, toxic substances and heavy metals. Exhaust emissions from road transport arise from the combustion of fuels such as gasoline, diesel, liquefied petroleum gas (LPG) and natural gas in internal combustion engines. The air/fuel charge may be ignited by a spark or it may ignite spontaneously when compressed.

The gas emission depends on several factors such as; type of the car, fuel used, engine size, weight, technology level of vehicle etc. It is also depends on the road types such as rural, urban or highway. The combustion process produces CO_2 and H_2O as the main products. Unfortunately, combustion also produces several by-products.

Given the diversity in the in propulsion concepts, the calculation of emissions from road vehicles is a complicated and demanding procedure which requires good quality activity data and emission factors [16]. In this study, it is assumed that all vehicles are the same type and the gas emission functions of all vehicles are identical. The emission function and parameters used in the model are obtained from the website of Copert 4 which is an Microsoft Windows software program aiming at the calculation of air pollutant emissions from road transport. The technical development of COPERT is financed by European Environment Agency in the framework of the activities of the European Topic Center on Air and Climate Change. Since 2007, European Commission's Joint Research Center has been coordinating the further scientific development of the model. In principle, COPERT has been developed for use from National Experts to estimate emissions from road transport to be included in official annual national inventories.

2.5 BI-LEVEL PROGRAMMING

Of the various types of mathematical two-level structures, the general bi-level programming problem is the most challenging. It was originally proposed as a model for a leader-follower game in which two players try to minimize their individual objective functions $F(x, y)$ and $f(x, y)$, respectively, subject to a series of interdependent constraints. Once again, the underlying assumptions are that full information is available, at least to the leader and that cooperation is prohibited. This precludes the use of correlated strategies and side payments [17].

The hierarchical relationship results from the fact that the mathematical program related to the users' behavior is part of the manager's constraints. This is the major feature of bi-level programs: they include two mathematical programs within a single instance, one of these problems being part of the constraints of other one. In view of this hierarchical relationship, the program can be separated into upper-level problem and lower-level problem [18].

The general formulation of bi-level programming problem is;

$$\begin{aligned} &\min_{x\in X,y} \quad F(x,y) \\ &s.t. \quad G(x,y) \le 0 \\ &\qquad \min_{y} \quad f(x,y) \\ &\qquad s.t. \quad g(x,y) \end{aligned} \tag{2.23}$$

where x $\in \mathbb{R}^{n_1}$ and y $\in \mathbb{R}^{n_2}$. The variables of the problem are divided in to two classes, namely the upper level variables x $\in \mathbb{R}^{n_1}$ and the lower level variables y $\in \mathbb{R}^{n_2}$. Similarly, the functions $F : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}$ and $f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}$ are the upper level and lower level objective functions respectively, while the vector valued functions $G : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{m_1}$ and $g : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{m_2}$ are called the upper level and lower level constraints respectively. Upper level constraints involve variables from both levels and play a very specific role. Indeed they must be enforced indirectly, as they do not bind the lower level decision maker [18].

The relaxed problem corresponding to (2.23) is;

$$\begin{aligned} &\min_{x\in X,y} F(x,y) \\ &s.t. \quad G(x,y) \le 0 \\ &\qquad g(x,y) \end{aligned} \tag{2.24}$$

and its optimal value is a lower bound for the optimal value of (2.23). The relaxed feasible region is

$$\Omega = \{(x,y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} : x \in X, G(x,y) \le 0 \ and \ g(x,y) \le 0\} \tag{2.25}$$

For a fixed vector $\bar{x} \in X$, the lower level feasible region is

$$\Omega(\bar{x}) = \{y \in \mathbb{R}^{n_2} : g(\bar{x},y) \le 0\} \tag{2.26}$$

while the lower reaction set is defined by

$$R(\bar{x}) = \{y \in \mathbb{R}^{n_2} : y \in argmin\{f(\bar{x}, \hat{y}) : \hat{y} \in \Omega(\bar{x})\}\} \quad (2.27)$$

Every $y \in R(\bar{x})$ is a rational response. For a given x, $R(x)$ is an implicitly defined multi-valued function of x that may be empty for some values of its argument. Finally the set;

$$IR = \{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} : x \in X, G(x, y) \leq 0, y \in R(x)\} \quad (2.28)$$

that regroups the feasible points of the BLPP, corresponding to the feasible set of the leader and is known as the induced region. This set is usually non-convex and it can even be disconnected or empty in presence of upper level constraints [18].

Both problems (lower and upper) are equivalent provided that the lower level problem is a convex one with a unique optimal solution and validity of a regularity assumption for all parameter values [19].

Being generically non-convex and non-differentiable, bi-level programs intrinsically hard. Even the simplest instance, the linear BLLP, was shown to be NP-hard by Jeroslaw et al [20]. Bi-level programming problem is generally a non-convex and non-differentiable optimization problem with implicitly determined objective and constraint functions [19].

Stackelberg Game

The Stackelberg game is a problem of mathematical game theory identical to the bi-level programming problem [19]. In the particular framework of Stackelberg games, the leader is assumed to anticipate the reaction of the followers; this allows him to choose his best –or optimal- strategy accordingly. More precisely, the leader chooses a strategy x in a set $X \subseteq \mathbb{R}^n$ and every follower i has a strategy set $Y_i(x) \subseteq \mathbb{R}^{m_i}$ corresponding to each $x \in X$. The sets $Y_i(x)$ are assumed to be closed and convex. In an oligopolistic situation, Stackelberg problems possess a hierarchical structure similar to that of BLPP, although the lower level problem is equilibrium rather than an optimization problem [18].

Many decision-making problems for transportation planning and management can be described as a Stackelberg game. In a Stackelberg game, the leader knows how the follower will respond to any decision he may make, that is, the system manager can influence but can not control travelers' choices. In the light of any control decision, travelers make their own travel choices, specifically, route choices in a user optimal manner. It is assumed that for any given control pattern, there is a unique equilibrium flow distribution. The equilibrium flow distribution is also called as response or reaction [21].

It is assumed that the travelers make their route choices in a user-optimal manner; hence the lower level problem can be formulated as a standard user equilibrium traffic assignment problem [1].

2.6 MULTI-OBJECTIVE PROGRAMMING

A multi-objective optimization problem (MOOP) deals with more than one objective function. In most practical decision-making problems, multiple objectives or multiple criteria are evident. Because of a lack of suitable solution methodologies, an MOOP has been mostly cast and solved as a single-objective optimization problem in the past. However, there exist a number of fundamental differences between the working principles of single and multi-objective optimization algorithms. In a single-objective optimization problem, the task is to find one solution (except in some specific multi-modal optimization problems where multiple optimal solutions are sought) which optimizes the sole objective function. Extending the idea to multi-objective optimization is to find an optimal solution corresponding to each objective function.

The general form of the MOOP [22]:

$$\begin{array}{llll} \min \setminus \max & f_m(x), & \mathrm{m} = 1, 2, \ldots, M; \\ subject\ to & g_j(x) \geq 0 & j = 1, 2, \ldots, J; \\ & h_k(x) = 0 & k = 1, 2, \ldots, K; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)} & i = 1, 2 \ldots, n. \end{array} \tag{2.29}$$

A solution x is a vector of n decision variables, $x = (x_1, x_2, \ldots, x_n)^T$. The last set of constraints is called variable bounds, restricting each decision variable x_i to take values within a lower $x_i^{(L)}$ and an upper $x_i^{(U)}$ bound. These bounds constitute a decision variable space D, or simply the decision space.

One of the striking differences between single-objective and multi-objective optimization is that in multi-objective optimization the objective functions constitute a multi-dimensional space, in addition to the usual decision variable.

Definition 2.1. *A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex function if for any two pair of solutions $x^{(1)}, x^{(2)} \in \mathbb{R}^n$, the following condition is true:*

$$f(\lambda x^{(1)} + (1-\lambda)x^{(2)}) \leq \lambda f(x^{(1)}) + (1-\lambda)f(x^{(2)}) \tag{2.30}$$

for all $0 \leq \lambda \leq 1$.

Definition 2.2. *A multi-objective optimization problem is convex if all objective functions are convex and the feasible region is convex (or all inequality constraints are non-convex and equality constraints are linear.*

According to this definition, a Multi-Objective Linear Programming is a convex problem. Since MOOPs has two spaces, the convexity in each space (objective and decision variable space) is important to multi-objective optimization algorithm.

Definition 2.3. *A solution* $x^{(1)}$ *is said to dominate the other solution* $x^{(2)}$*, if both conditions 1 and 2 are true:*

1. *The solution* $x^{(1)}$ *is no worse than* $x^{(2)}$ *in all objectives.*
2. *The solution* $x^{(1)}$ *is strictly better than* $x^{(2)}$ *in at least one objective* $\bar{j} \in \{1, 2, \ldots, M\}$.

It is intuitive that if a solution $x^{(1)}$ dominates another solution $x^{(2)}$, the solution $x^{(1)}$ is better than $x^{(2)}$ in the parlance of multi-objective optimization.

Definition 2.4 (Strong dominance). *A solution* $x^{(1)}$ *strongly dominates a solution* $x^{(2)}$*, if solution* $x^{(1)}$ *is strictly better than solution* $x^{(2)}$ *in all* M *objectives.*

Definition 2.5 (Weekly non-dominated set). *Among a set of solutions* P*, the weekly non-dominated set of solutions* P' *are those that are not strongly dominated by any other member of the set* P.

If two solutions, $x^{(1)}$ and $x^{(2)}$, can not dominate each other, it is customary to say that solutions are non-dominated with respect to each other. When both objectives are important, it can not be said which of these solutions is better.

Definition 2.6 (Non-dominated set). *Among a set of solutions* P*, the non-dominated set of solutions* P'*are those that are not dominated by any member of the set* P.

When the set P is the entire search space, or $P = S$, the resulting non-dominated set P' is called the Pareto-optimal set. It should be noted that although the search space can be non-convex, the Pareto-optimal front may be convex.

From definitions, it is clear that a Pareto-optimal set is always a non-dominated set. But there may exist non-dominated sets containing some Pareto-optimal solutions and some non-Pareto-optimal solutions. Thus, it is important to realize that the non-dominated solutions found by an optimization algorithm need not represent the true Pareto-optimal set [22].

In the case of conflicting objectives, usually the set of optimal solutions contains more than one solution. In the presence of multiple Pareto-optimal solutions, it is difficult to prefer one solution over the other without any further information about the problem. However, in the absence of any such information, all Pareto-optimal solutions are equally important. In the light of ideal approach, it is important to find as many Pareto-optimal solutions as possible in a problem. Thus, it can be conjectured that there are two goals in a multi-objective optimization.

1. To find a set of solutions as close as possible to the Pareto-optimal front.
2. To find a set of solutions as diverse as possible.

The first goal is mandatory in any optimization task. Converging to a set of solutions which are not close to the true optimal set of solutions is not desirable. This goal of multi-objective optimization is common to the similar optimality goal in the single-objective optimization.

On the other hand, the second goal is entirely specific to multi-objective optimization. In addition to being converged close to the Pareto-optimal front, they must be sparsely spaced in the Pareto-optimal region. Only with a diverse set of solutions, having a good set of trade-off solutions among objectives can be assured [22].

3 MATHEMATICAL MODEL

The rapid transit network design (RTND) problem defined in [11] finds the optimal lines and station, considering simultaneously the optimal user route and mode. For given possible stations, it studies the problem of selecting and then connecting them with a small number of RTN lines $L = \{l = 1, 2, \ldots, |L|\}$ in competition with the rest of the modes, that is considered by the alternative mode [14]. The aim of this model is minimizing the construction cost and emission raised from the vehicles and at the same time keeping the network at equilibrium that minimizes total travel time.

3.1 ASSUMPTIONS

The assumptions made in this study are;

- Travel times depend on the number of vehicles on each link.
- All users have perfect information about the travel times in each link.
- OD matrix is used instead of corridor approach and catchment areas.
- A single line with no predetermined origin destination nodes is to be constructed.
- Users respond to the changes in the system rationally. In other words, users will select the mode or the route with the shortest traveling time.
- Vehicles are identical, they are the same model, at the same age and use gasoline for combustion and occupying factor (capacity) of each vehicle is 1.
- A user can use transit line if and only if both of his/her origin and destination nodes are included by the alignment.
- It is assumed that links with shorter distances last less than the longer ones.

3.2 NOTATION

Sets

A : set of links in the existing network

$\hat{A}$: set of possible links in the transit network where $|\hat{A}| = N(N-1)$

N : set of centroids and stations

$N(i)$: set of nodes adjacent to node$i$

$\hat{N}$: set of nodes in which stations are located where $\hat{N} = \{i : y_i = 1\}$

Variables

x_{ij}^{rs} : flow on link between node i and node j originating from origin r to destination s

$\hat{x}_{ij}^{rs}$: flow on transit link between node i and node j originating from origin r to destination s

$\hat{d}_{rs}$: demand using transit network originating from origin r to destination s

d_{rs} : demand using existing network originating from origin r to destination s

f_{ij} : total flow using link between node i and node j

$\hat{f}_{ij}$: flow using transit link between node i and node j

Ψ_{rs} : perceived time difference between transit and road networks for users traveling from origin r to destination s

$v_{ij}(x_{ij})$: average vehicle speed on link (i,j) given flow x_{ij}

e_{ij} : unit vehicle emission on link (i,j) given average vehicle speed v_{ij}

$$u_{ij} = \begin{cases} 1, & \text{if line is located using the link between node } i \text{ and node } j \\ 0, & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{if a station is located in node } i \\ 0, & \text{otherwise} \end{cases}$$

$Parameters$

$\bar{d}_{rs}$: total demand originating from origin r to destination s

θ : a parameter that is calibrated from observations regarding users' choices of destination

t_{ij}^{0} : free flow time in the link (i, j)

α : a constant used in (2.1) with value 0,15.

β : a constant used in (2.1) with value 4.

h_{ij} : cost of connecting the links i and j with transit network

l_{ij} : length of the transit link between nodes i and j

c_{ij} : capacity of the link (i, j)

3.3 CONSTRAINTS

3.3.1 Upper-Level Constraints

$$u_{ij} \leq y_i \qquad (i, j) \in |\hat{A}| \qquad (3.1)$$

Constraint (6.1) ensures that a transit link originating from node i to node j does not exist if a station is not located at node i.

$$u_{ij} \leq y_j \qquad (i, j) \in |\hat{A}| \qquad (3.2)$$

Constraint (6.2) states that a transit link does not exist from node i to node j if a station is not located at node j.

$$u_{ij} = u_{ji} \qquad (i, j) \in |\hat{A}| \qquad (3.3)$$

In Constraint (6.3), it is shown that transit links are unidirectional, that means it is possible to travel in both ways.

$$\sum_{j \in N(i),\, i<j} u_{ij} + \sum_{j \in N(i),\, j<i} u_{ji} \leq 2 \qquad i \in N \qquad (3.4)$$

Constraint (6.4) dictates that the number of incoming and outgoing links from node j could not exceed 2. For only origin and terminus nodes of the alignment, this value could be 1 and it will be 0 if a station is not located at node i.

$$1 + \sum_{(i,j)\in\hat{A},\ i<j} u_{ij} = \sum_{i\in N} y_i \tag{3.5}$$

Constraint (6.5) forces that the number of stations in a line must exceed the number of links connecting these stations by one.

$$\sum_{i\in B} \sum_{j\in N} u_{ij} \leq |B| - 1 \qquad \forall B \subset N,\ |B| \geq 2 \tag{3.6}$$

Possible sub-tours are prevented by Constraint (6.6) which is needed in the model to ensure that there is only one line to be constructed.

3.3.2 Lower-Level Constraints

$$\sum_{j:(i,j)\in A} x_{ij}^{rs} - \sum_{j:(j,i)\in A} x_{ji}^{rs} = \begin{cases} \bar{d}_{rs} - \hat{d}_{rs}, & i = r \\ 0, & i \neq r,\ s \\ -\bar{d}_{rs} + \hat{d}_{rs}, & i = s \end{cases} \qquad i \in N, \forall r, s \tag{3.7}$$

Constraint (6.7) ensures that node i is the origin of the $r - s$ pair if the number of outgoing flow from this node, which is traveling between r and s, is more than the number of incoming flow to this node which is traveling between r and s. Node i is destination of this $r - s$ pair if the number of incoming flow to this node, which is traveling between r and s, is more than the number of outgoing flow from this node, which is traveling between r and s. If they are equal, node i is neither origin nor destination of this pair. Here, $\bar{d}_{rs} - \hat{d}_{rs}$ represents the number of users traveling by their vehicles.

$$x_{ij}^{rs} \geq 0 \qquad (i,j) \in A, \forall r, s \tag{3.8}$$

It is shown that flows between any $r - s$ pair using the link (i, j) must be non-negative in Constraint (6.8).

$$f_{ij} = \sum_{rs} x_{ij}^{rs} \qquad (i,j) \in A \qquad (3.9)$$

In Constraint (6.9), it is shown that the total flow on any link is obtained by the sum of flows of all OD pairs passing through that link.

$$\sum_{j:(i,j)\in\hat{A}} \hat{x}_{ij}^{rs} - \sum_{j:(j,i)\in\hat{A}} \hat{x}_{ji}^{rs} = \begin{cases} \hat{d}_{rs}, & i = r \\ 0, & i \neq r, s \\ -\hat{d}_{rs}, & i = s \end{cases} \qquad i \in \hat{N}, \forall r, s \qquad (3.10)$$

Similar to (6.7), Constraint (6.8) ensures that node i is the origin of the $r - s$ pair if the number of outgoing flows from this node, which is traveling between r and s by transit line, is more than the number of incoming flow to this node, which is traveling between r and s by transit line. Node i is destination of this $r - s$ pair if the number of incoming flow to this node, which is traveling between r and s by transit line, is more than the number of outgoing flow from this node, which is traveling between r and s by transit line. If they are equal, node i is neither origin nor destination of this pair. As mentioned before, $\hat{d}_{rs}$ represents the number of users traveling by transit line.

$$0 \leq \hat{x}_{ij}^{rs} \leq M u_{ij} \qquad (i,j) \in \hat{A}, \forall r, s \qquad (3.11)$$

Constraint (6.11) ensures that if the link (i, j) is not included in the transit line, then no flow can be assigned to it.

$$\hat{d}_{rs} \leq M y_r \qquad \forall r, s \qquad (3.12)$$

In Constraint (6.12) it is assured that if a station is not located at origin r, then flows originating from this node can not use transit line.

$$\hat{d}_{rs} \leq My_s \qquad \forall r, s \qquad (3.13)$$

In a similar manner to the (6.12), Constraint (6.13) shows that if a station is not located at destination s, flows destined to this node can not use transit line.

3.4 OBJECTIVE FUNCTIONS

Before presenting the objective functions, it will be useful to give information about gas emission objective function. The parameters and the function are obtained from [14] which is available in the website of the Copert 4 Software. Because CO has the biggest portion in all of the harmful gases released from the vehicles, parameters corresponding to it are used in the formula. This formula is derived for a EURO 3 type vehicle which uses gasoline for combustion. Since extraction of the formula in the objective function and the meanings of the parameters are out of scope of this study, only necessary information is given.

However the parameter V, which is the average speed of the vehicles in the network, is closely related to our formulation. As mentioned earlier, travel time of one vehicle in a given link is computed by using (2.1). Since the length of any link is the product of the average speed and the travel time on that link, it can be easily figured out what the average speed of the vehicles is because time and length values are available. Given the number of vehicles in the link (i, j), average speed of one vehicle in terms of (km\h) is;

$$v_{ij}(x_{ij}) : \frac{l_{ij}}{t_{ij} = t^0_{ij}\left[1 + \alpha\left(\frac{x_{ij}}{c_{ij}}\right)^{\beta}\right]} \qquad (3.14)$$

Given the average vehicle speed in link (i, j), gas emission per vehicle in terms of (g\km), is computed by

$$e_{ij}(v_{ij}) : \frac{(a + c \times v_{ij})}{(1 + b \times v_{ij} + d \times v^2_{ij})} \qquad (3.15)$$

where

$$a = 7.17 \times 10^1$$
$$b = 3.54 \times 10^1$$
$$c = 1.14 \times 10^1$$
$$d = -2.48 \times 10^{-1}.$$

$$\min \sum_{(i,j)\in A} e_{ij} l_{ij} x_{ij} \tag{3.16}$$

$$\min \sum_{i\in\hat{N}} \sum_{j\in\hat{N}} h_{ij} u_{ij} \tag{3.17}$$

(6.16) and (6.17) are the objective functions of the upper level problem in our model. The first one is minimizing the gas emission in the entire network while the second one is the construction cost minimizing objective function.

$$\min \quad \sum_{(i,j)\in A} \int_0^{f_{ij}} t_{ij}(w)dw + \sum_{rs} \int_0^{\hat{d}_{rs}} \frac{1}{\theta} ln(\frac{w}{\bar{d}_{rs} - w} + \Psi_{rs})dw \tag{3.18}$$

(6.18) is the objective function of the lower level problem. This function aims to minimize travel time of the users. The first term is the total travel time spent by users traveling by their vehicles. Likewise, the second term is the travel time spent by users traveling by transit line. At the point where the sum of these terms is minimized, the model finds user equilibrium which travel times of all paths and all modes are the same.

4 SOLUTION METHODOLOGY

4.1 NON-DOMINATED SORTING GENETIC ALGORITHM-II

Deb et al [23] suggested an elitist non-dominated sorting genetic algorithm, termed NSGA-II. NSGA-II uses an explicit diversity-preserving mechanism instead of using only an elite-preservation mechanism.

At the beginning of NSGA-II, for each solution two entities are calculated, domination count n_p, the number of solutions which dominate the solution p and S_p, a set of solutions that the solution p dominates. All solutions in the first non-dominated front will have their non-domination count as zero. For each solution p with $n_p = 0$, each member (q) of its set S_p is visited and its domination count is reduced by one. In doing so, if for any member q, the domination count becomes zero, it is put in a separate list Q. These members belong to the second non-dominated front. The above procedure is continued with each member of Q and the third front is identified. This process continues until all front are identified [23]. The procedure of fast non-dominated sort is outlined below:

Step 1. $S_p = \emptyset$ and $n_p = 0$, $\forall p \in P$.

Step 2. $\forall q \in P$, if p dominates q, $S_p = S_p \cup \{q\}$, else $n_p = n_p + 1$.

Step 3. If $n_p = 0$, $p_{rank} = 1$. $F_1 = F_1 + \{p\}$.

Step 4. Initialize $i = 1$. While $F_i = \emptyset$, $Q = \emptyset$. $n_q = n_q - 1$ $\forall p \in F_i$ and $\forall q \in S_p$. If $n_q = 0$, $q_rank = 1$ and $Q = Q + \{q\}$. Set $i = i + 1$ and $F_i = Q$.

In NSGA-II, the offspring population Q_t is first created by using the parent population P_t. However, instead of finding the non-dominated front of Q_t only, the first two populations are combined together to form R_t of size $2N$. Then, a non-dominated sorting is used to classify the entire population R_t. Although, this requires more effort compared performing a non-dominated sorting on Q_t alone, it allows a global non-domination check among the offspring and parent solutions. Once the non-dominated sorting is over, the new population is filled by solutions of different non-dominated front and continues with solutions of the second non-dominated fronts, one at a time. The filling starts with the best non-dominated front and continues with solutions of the second non-dominated front, followed by the third non-dominated front, and so on. Since the overall population size of R_t is $2N$, not all fronts may be accommodated in N slots available in the new population. All fronts which could not be accommodated are simply deleted. When the last allowed front is being considered, there may exist more solutions in the last front than the remaining slots in the new population.

A strategy like the above does not affect the proceeding of the algorithm much in the early stages of the evolution. This is because, early on, there exist many fronts in the combined population. It is likely that solutions of many good non-dominated fronts are already included in the new population, before they add up to N. It then hardly matters which solution is included to fill up the population. However, during the latter stages of the simulation, it is likely that most solutions in the population lie in the best non-dominated front. It is also likely that in the combined population R_t of size $2N$, the number of solutions in the first non-dominated front exceeds N. The above algorithm then ensures that niching will choose a diverse set of solutions from this set. When the entire population converges to the Pareto-optimal front, the continuation of this algorithm will ensure a better spread among the solutions [22].

Step 1. Combine parent and offspring populations and create $R_t = P_t \cup Q_t$. Perform a non-dominated sorting to R_t and identify different fronts: $F_i = 1, 2, \ldots$, etc.

Step 2. Set new population. Set $i = 1$.

Until $|P_{t+1}| + |F| < N$, perform $P_{t+1} = P_{t+1} \cup F_i$ and $i = i + 1$.

Step 3. Perform the Crowding-sort $(F_i, <_c)$ procedure and include the most widely spread $(N - |P_{t+1}|)$ solutions by using the crowding distance values in the sorted F_i to P_{t+1}.

Step 4. Create offspring population Q_{t+1} from P_{t+1} using the crowded tournament selection, crossover and mutation operators.

Crowded Tournament Selection Operator

Definition 3.1 *Crowded Tournament Selection Operator: A solution i wins a tournament with another solution j if any of the following conditions are true:*

1. *If solution i has a better rank, that is $r_i < r_j$.*
2. *If they have the same rank but solution i has a better crowding distance than solution j, that is, $r_i = r_j$ and $d_i > d_j$.*

Crowding Distance

To get an estimate of the density of solutions surrounding a particular solution i in the population, the average distance of two solutions on either side of solution i along each objective is computed. This quantity d_i serves as an estimate of the perimeter of the cuboid formed by using nearest neighbors as the vertices. Crowding distance assignment procedure $(F, <_c)$, is as follows:

Step 1. Call the number of solutions in F as $l = |F|$. For each i in the set, first assign $d_i = 0$.

Step 2. For each objective function $m = 1, 2, \dots, M$, sort the set in worse order of f_m or, find the sorted indices vector: $I^m = sort(f_m, >)$.

Step 3. For $m = 1, 2, \dots, M$, assign a large distance to the boundary solutions, or $d_{I_l^m} = d_{I_l^m} = \infty$, and for all other solutions $j = 2$ to $(l - 1)$, assign:

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{max} - f_m^{min}}. \tag{4.1}$$

Advantages

Since solutions compete with their crowding distances, no extra niching parameter is required in NSGA_II. Furthermore, the elitism mechanism does not allow an already found Pareto-optimal solution to be deleted.

Disadvantages

As long as the size of the first non-dominated set is not larger than the population size, the algorithm preserves all of them. However, in latter generations, when more than N members belong to the first non-dominated set in the combined parent-offspring population, some closely-packed Pareto-optimal solutions may give their places to other non-dominated yet non-Pareto-optimal solutions.

4.2 OTHER EVOLUTIONARY ALGORITHMS FOR MOOP

4.2.1 Non-Elitist Multi-Objective Evolutionary Algorithms

4.2.1.1 Vector Evaluated Genetic Algorithm

Schaffer et al [24] implemented the first multi-objective GA to find a set of non-dominated solutions. Subsequently, he compared his GA with an adaptive random search technique in [25] and observed a better performance obtained by his algorithm. The name is appropriate for multi-objective optimization because the algorithm evaluates an objective vector (instead of a scalar objective function) with each element of the vector representing each objective function.

Vector Evaluated Genetic Algorithm (VEGA) is the simplest possible multi-objective Genetic Algorithm (GA) and is a straightforward extension of a single objective GA. Since a number of objectives (M) have to be handled, GA population is divided at every iteration into M equal subpopulations randomly. Vector Evaluated Genetic Algorithm procedure is as follows:

Step 1. Set an objective function counter $i = 1$ and define $q = N/M$.

Step 2. For all solutions $j = 1 + (i-1) * q$ to $j = i * q$, assign fitness a:

$$F(x^{(j)}) = f_i(x^{(j)}) \tag{4.2}$$

Step 3. If $i = M$, go to Step 5. Otherwise, increment i by one and go to Step 2.

Step 4. Combine all mating pools together: $P = \cup_{i=1}^{M} P_i$. Perform crossover an mutation on P to create a new population.

In order to find intermediate trade-off solutions, crossover between any two solutions in the entire population is allowed. In this way, it is aimed to find good compromised solutions between the objectives.

Advantages

The main advantage of VEGA is the ease of implementing. Only minor changes are required to be made in a simple GA to convert it to a multi-objective GA and this does not incur any additional computational complexity

Disadvantages

Because each solution in a VEGA is evaluated with only one objective function, it is likely that solutions near the optimum of an individual objective function would be preferred by the selection operator in a subpopulation. Eventually, the VEGA converges to individual champion solutions only.

4.2.1.2 Non-Dominated Sorting Genetic Algorithm

The idea of using the non-dominated sorting concept in GAs of Goldberg et al [26] was more directly implemented by Srinivas et al [27]. The dual objectives in a multi-objective optimization algorithm are maintained by using a fitness assignment scheme which prefers non-dominated solutions and by using a sharing strategy which preserves diversity among solutions of each non-dominated front.

The first step of Non-Dominated Sorted Genetic Algorithm (NSGA) is to sort the population according to non-domination. This classifies the population into a number of mutually exclusive equivalent classes P_j:

$$P = \cup_{j=1}^{\rho} P_j \tag{4.3}$$

Any two members from the same class can not be said to be better than one another with respect to all objectives. The total number of classes (fronts), denoted as ρ, in the above equation, depends on the population P and the underlying problem.

Once the classification is over, it is obvious that all solutions in the first front, that is, all $i \in P_1$, belong to the best non-dominated set in the population. That means, the solutions in the first front, are best in terms of their closeness to the true Pareto-optimal front in the population.

The fitness assignment procedure begins with the first non-dominated set with the highest fitness and successively proceeds to dominated sets. Assignment more fitness to solutions belonging to a better non-dominated set ensures a selection pressure towards the Pareto-optimal front. However, in order to achieve the second goal, diversity among solutions in a front must also be maintained. In NSGA, the diversity is maintained by degrading the assigned fitness based on the number of neighboring solutions. Diversity is important because it provides a better representation of solutions among the Pareto-optimal set.

The sharing function method is used front-wise. That is, for each solution i in the front P_1, the normalized Euclidean distance d_{ij} from another solution j in the same front is calculated. Once these distances are calculated, they are used to compute a sharing function. The sharing function takes a value between zero and one, depending on the distance d_{ij}. After all sharing function values are calculated, they are added together to calculate the niche count ηc_i of the i^{th} solution. The niche count denotes the number of solutions in the neighborhood of the i^{th} solution. The process of degrading fitness of a solution which is crowded by many solutions helps emphasize the solutions residing in less crowded regions.

The sharing function method works with the proportionate selection operator and this assigns copies in the mating pool proportional to the shared fitness. In this way, each solution in the first front has a better chance of surviving in the mating pool than in the second front, and so on.

Advantages

The main advantage of NSGA is the assignment of fitness according to non-dominated sets. Since better non-dominated sets are emphasized systematically, an NSGA progresses towards the Pareto-optimal region front-wise. Moreover, performing sharing in the parameter space allows phenotypically diverse solutions to emerge when using NSGAs.

Disadvantages

The sharing function approach requires fixing the parameter σ_{share}. It has been observed earlier that the performance of an NSGA is sensitive to this parameter.

4.2.1.3 Niched –Pareto Genetic Algorithm

The Niched Pareto Genetic Algorithm (NPGA) proposed by Horn et al [28] uses the binary tournament selection operator, unlike the proportionate selection methods in other algorithms. It has been shown that the tournament selection has better growth and convergence properties compared to the proportionate selection. However, when a tournament selection is to be applied to multi-objective GAs which use the sharing approach, it is necessary to use the proportionate selection operator.

NPGA Tournament Selection Procedure

winner=NPGA-tournament(i, j, Q)

Step 1. Pick a subpopulation T_{ij} of size t_{dom} from the parent population P.

Step 2. Find α_i as the number of solutions in T_{ij} that dominates i. Calculate α_j as the number of solutions in T_{ij} that dominates j.

Step 3. If $\alpha_i = 0$ and $\alpha_j > 0$, then i is the winner. The process is complete.
Step 4. Otherwise, if $\alpha_i > 0$ and $\alpha_j = 0$, then j is the winner. The process is complete.

Step 5. Otherwise, current offspring population, $|Q| < 2$, i and j is chosen as the winner with probability 0.5. The process is complete. Alternatively, the niche counts ηc_i and ηc_j are calculated by placing i and j in the current offspring population Q, independently. With the niching parameter σ_{share}, ηc_i is calculated as the number of offspring $(k \in Q)$ within a σ_{share} distance d_{ik} from i. The distance d_{ik} is the Euclidean distance between solutions i and k in the objective space.

Step 6. If $\eta c_i < \eta c_j$, solution i is the winner. Otherwise, solution j is the winner.

NPGA Procedure

Step 1. Shuffle P, set $i = 1$ and set $Q = \emptyset$.

Step 2. Perform tournament selection and find the first parent, $p_1 =$ NPGA-tournament(i,i+1,Q).

Step 3. Set $i = i + 2$ and find the second parent, $p_2 =$ NPGA-tournament(i,i+1,Q).

Step 4. Perform crossover with p_1 and p_2 create offspring c_1 and c_2. Perform mutation on c_1 and c_2.

Step 5. Update offspring population $Q = Q \cup \{c_1, c_2\}$.

Step 6. Set $i = i + 1$. If $i < N$, go to Step 2. Otherwise, if $|Q| = N/2$, shuffle P, set $i = 1$ and go to Step 2. Otherwise, process is complete.

Advantages

One of the main advantages of the NPGA is that no explicit fitness assignment is needed. Another advantage of the NPGA is that this is the first proposed multi-objective evolutionary algorithm which uses the tournament selection operator.

Disadvantages

The NPGA requires fixing two important parameters: σ_{share} and t_{dom}. In addition, σ_{share} parameter has more effect on an NPGA than NSGA.

4.2.2 Elitist Multi-Objective Evolutionary Algorithms

4.2.2.1 Strength Pareto Evolutionary Algorithm

This algorithm which is proposed by Zitzler et al [29] introduces elitism by explicitly maintaining an external population $\bar{P}$. This population stores a fixed number of the non-dominated solutions that are found until beginning of a simulation. At every generation, newly found non-dominated solutions are compared with the existing external population and the resulting non-dominated solutions are preserved. The Strength Pareto Evolutionary Algorithm (SPEA) does more than just preserving the elites; it also uses these elites to participate in the genetic operations along with the current population in the hope of influencing the population to steer towards good regions in the search space.

Strength Pareto Evolutionary Algorithm

Step 1. Find the best non-dominated set $F_1(P_t)$ of P_t. Copy these solutions to $\bar{P}_t$, or perform $\bar{P}_t = \bar{P} \cup F_1(P_t)$.

Step 2. Find the best non-dominated solutions $F_1(\bar{P}_t)$ of the modified population $\bar{P}_t$ and delete all dominated solutions, or perform $\bar{P}_t = F_1(\bar{P}_t)$.

Step 3. If $|\bar{P}_t| > N$, use clustering technique to reduce the size to $\bar{N}$. Otherwise, keep $\bar{P}_t$ unchanged. The resulting population is the external population $\bar{P}_{t+1}$ of the next generation.

Step 4. Assign fitness to each elite solution $i \in \bar{P}_{t+1}$ by using equation $S_i = \dfrac{n_i}{N+1}$. Then, assign fitness to each population member $j \in P_t$ by using equation;

$$F_j = 1 + \sum_{i \in \bar{P}_t \wedge i \preceq j} S_i \tag{4.4}$$

Step 5. Apply a binary tournament selection with these fitness values, a crossover and a mutation operator to create the new population P_{t+1} of size N from the combined population $(\bar{P}_{t+1} \cup P_t)$ of size $(\bar{N} + N)$.

Clustering algorithm is used when the size of the external population $\bar{P}_t$ is more than the fixed external population size $\bar{N}$ $(\bar{N}' > \bar{N})$.

Clustering Algorithm

Step 1. Initially, each solution belongs to the distinct cluster or $C_i = \{i\}$, so that $C = \{C_1, C_2, \ldots, C_{\bar{N}'}\}$.

Step 2. If $|C| \leq \bar{N}$, go to Step 5. Otherwise, go to Step 3.

Step 3. For each pair of clusters, calculate the cluster-distance by using equation

$$d_{12} = \frac{1}{|C_1||C_2|}| = \sum_{i \in C_1, j \in C_2} d(i, j) \tag{4.5}$$

Find the pair (i_1, i_2) which corresponds to the minimum cluster-distance.

Step 4. Merge two clusters C_{i_1} and C_{i_2} together. This reduces the size of C by one. Go to Step 2.

Step 5. Choose only one solution from each cluster and remove the others from the clusters. The solution having the minimum average distance from other solutions in the cluster can be chosen as the representative solution of a cluster.

Advantages

Once a solution in the Pareto-optimal front is found, it immediately gets stored in the external population. The only way it gets eliminated is when another Pareto-optimal solution, which leads to a better spread in the Pareto-optimal solutions, is discovered. Clustering algorithm which is parameter-less, provides a better spread of the obtained non-dominated solutions.

Disadvantages

The SPEA introduces an extra parameter $\bar{N}$, the size of external population. A balance between the regular population size N and this external population size $\bar{N}$ is important in the successful working of the SPEA.

Since clustering algorithm has an $O(MN^2)$ complexity which is larger than the crowding strategy used in NSGA-II. Thus, an SPEA's niche-preservation operator can be made faster by using crowded strategy.

Since non-dominated sorting of the whole population is not used for assigning fitness, the fitness values do not favor all non-dominated solutions of the some rank equally. This bias in fitness assignment in the solutions of the same front is dependent on the exact population and densities of solutions in the search space.

Moreover, in the SPEA fitness assignment, an external solution which dominates more solutions gets a worse fitness. This assignment is justified when all dominated solutions are concentrated near the dominating solution. Since in most cases this is not true, the crowding effect should come only from the clustering procedure. Otherwise, this fitness assignment may provide a wrong selection pressure for the non-dominated solutions.

4.2.2.2 Pareto-Archived Evolution Strategy

Knowles et al [30] suggested a multi-objective evolutionary algorithm which uses an evolution strategy. In simplest form, the Pareto-archived Evolution Strategy (PAES) uses a $(1+1)-ES$. The main motivation for using an ES came from their experience in solving real world telecommunications network design problem. In the single-objective version of the network design problem, they observed that a local search strategy (such as simulated annealing and tabu search method) worked better than a population-based approach. Motivated by this fact, they investigated whether a multi-objective evolutionary algorithm with a local search strategy can be developed to solve the multi-objective version of the telecommunications network design problems. Since a $(1+1)-ES$ uses only mutation on a single parent to create a single offspring, this is a local search strategy, and thus the investigators developed their first multi-objective evolutionary algorithm using $(1+1)-ES$.

Archive and Parent Update in PAES

Step 1. If c_t (the offspring) is dominated by any member of A_t (the archive), set $p_{t+1} = p_t$ (A_t is not updated). Process is complete. Otherwise, if c_t dominates a set of members from $A_t : D(c_t) = \{i : i \in A_t \wedge c_t \preceq i\}$, perform the following steps:

$$A_t = A_t \backslash D(c_t), \tag{4.6}$$

$$A_t = A_t \cup \{c_t\}, \tag{4.7}$$

$$p_{t+1} = c_t. \tag{4.8}$$

Process is complete. Otherwise, go to Step 2.

Step 2. Count the number of archived solutions in each hypercube (the entire search space is divided into $(2^d)^M$ unique, equal-sized M-dimensional hypercubes). The parent p_t belongs to a hypercube having n_p solutions, while the offspring belongs to a hypercube having n_c solutions. The highest count hypercube contains the maximum number of archived solutions.

If $|A_t| < N$(maximum number of archive), include the offspring in the archive, or $A_t = A_t \cup \{c_t\}$ and $p_{t+1} = Winner(c_t, p_t)$, and return. Otherwise (that is if $|A_t| = N$), check if c_t belongs to the highest-count hypercube. If yes, reject c_t, set $p_{t+1} = p$, and return. Otherwise, replace a random solution r from the highest-count hypercube with c_t:

$$A_t = A_t \backslash \{r\}, \tag{4.9}$$

$$A_t = A_t \cup \{c_t\}, \tag{4.10}$$

$$p_{t+1} = Winner(c_t, p_t). \tag{4.11}$$

The process is complete. The $Winner(c_t, p_t)$ chooses c_t, if $n_c < n_p$. Otherwise, it chooses p_t. It is important to note that in any parent-offspring scenario, only one of the above will be invoked.

Advantages

The PAES has a direct control on the diversity that can be achieved in the Pareto-optimal solutions. Step 2 of the algorithm emphasizes the less populated the hypercubes to survive, thereby ensuring the diversity.

Furthermore, since equal-sized hypercubes are chosen, the PAES should perform better when compared to other methods in handling problems having a search space with non-uniformly dense solutions.

Disadvantages

In addition to choosing an appropriate archive size N, the depth parameter d, which directly controls the hypercube size, is also an important parameter. A change of d changes the number of hypercubes exponentially, thereby making it difficult to arbitrarily control the spread solutions.

There is another difficulty with the PAES. Since the sizing of the hypercubes is performed with the minimum and the maximum bounds of the entire search space, when solutions converge near the Pareto-optimal front, the hypercubes are comparatively large.

4.2.2.3 Multi-objective Messy Genetic Algorithm

Veldhuizen et al [31], in his doctoral dissertation, proposed an entirely different approach to multi-objective optimization. He modified the single-objective messy GAs of Goldberg et al. [32] to find multiple Pareto-optimal solutions in multi-objective optimization.

Single-Objective Messy GAs

The main motivation in this study of messy GAs was to sequentially solve the two main issues of identifying salient building blocks in a problem and then combining the building blocks together to form the optimal or near-optimal solution. Messy GAs were successful in solving problems which can be decomposed into a number of overlapping or non-overlapping building blocks (partial solutions corresponding to the true optimal solution). Since the genic combination is an important matter to be discovered in a problem, messy GAs use both genic and allelic information in a string.

The above tasks of identification and combination of building blocks are achieved in two phases, (i) the primordial phase and (ii) the juxtapositional phase. In the primordial phase, the main focus is to identify and maintain the salient building blocks of a certain maximum order k. After the partial string is embedded in a template and is evaluated, a binary tournament selection with a niching approach is used to emphasize the salient partial strings in the primordial phase. Since the salient building constitute only a small fraction of all primordial strings, a systematic reduction in population sizing is also used. After a fair number of repetitive applications of tournament selection followed by a population reduction, the primordial phase is terminated.

In the juxtapositional phase, the salient building blocks are allowed to combine together with the help of a cut-and-splice and a mutation operator. The cut-and-splice operator is similar in principle to the single-point crossover, except that the cross sites in both parents need not fall at the same place. This causes variable-length chromosomes to exist in the population. The purpose of tournament selection with thresholding, cur-and-splice and the mutation operator is to find better and bigger building blocks, eventually leading to the true optimal solution.

The issue of using a locally optimal template string is crucial one. In the level-1 era of an mGA, all order one $(k = 1)$ substrings are initialized and evaluated with a template string created at random. At the end of this era, the obtained best solution is saved and used as a template for the level-2 era. In this era, all order two $(k = 2)$ substrings are initialized. This process is continued until a specified number of eras have elapsed or a specific number of function evaluations have been exceeded.

Modification for Multi-Objective Optimization

The primordial phase in multi-objective form of mGA, begins with a population identical to the single-objective mGAs. However, a different tournament selection procedure is used. The niched tournament selection operator is exactly the same as that in the NPGA approach. In order to compare two solutions in a population, a set of t_{dom} solutions are chosen from the latter. If one solution is non-dominated with respect to the chosen set and the other is not, the former solution is selected. On the other hand, if neither of them or both of them are terminated in the chosen set, a niching method is used to find which of the two solutions resides in a least crowded area. For both solutions, the niche count is calculated by using the sharing function approach with a predefined σ_{share} value. The solution with a smaller niche count is selected. The preference of non-dominated and less-crowded solutions using the above tournament selection helps to provide a selection advantage towards the non-dominated solutions and simultaneously maintains a diverse set of solutions. In all simulations, the investigator used the phenotypic sharing approach. The population reduction procedure is the same as that used in the single-objective mGAs.

The juxtapositional phase is also similar to that in the original mGA study, except that the above-mentioned niched tournament selection procedure is used. At the end of the juxtapositional phase, the current population is combined with an external population which stores a specified number of non-dominated solutions found thus far. Before ending this era, the external population is replaced with non-dominated solutions of the combined population. This dynamically updated procedure introduces elitism, which is an important property in any evolutionary algorithm. At the end of all MOMGA eras, the external set is reported as the obtained non-dominated set of solutions.

4.3 FRANK-WOLFE ALGORITHM

Frank-Wolfe Algorithm (also known as convex combinations method) was suggested by Frank and Wolfe as a procedure for solving quadratic programming problems with linear constraints [1].

It will be useful to mention about all-or-nothing assignment here briefly. In each iteration, travel times on each link and accordingly, total travel time of each path changes and all flow for a given OD pair $r - s$, q_{rs}, is assigned to the minimum-travel-time path connecting this pair. All other paths connecting this OD pair do not carry flow. This procedure is known as “all-or-nothing” assignment.

Step 0 ***Initialization.*** Perform all-or-nothing assignment based on $t_a = t_a(0),\ \forall a$. This yields $\{x_a^1\}$. Set counter $n = 1$.
Step 1 ***Update.*** Set $t_a^n = t_a(x_a^n),\ \forall n$.
Step 2 ***Direction finding.*** Perform all-or-nothing (AON) assignment based on $\{t_a^n\}$. This yields a set of auxiliary flows $\{y_a^n\}$.
Step 3 ***Line search.*** Find α_n that solves $\min\limits_{0 \leq \alpha \leq 1} \sum\limits_a \int_0^{x_n^a + \alpha(y_n^a - x_n^a)} t_a(w)dw$
Step 4 ***Move.*** Set $x_a^{n+1} = x_a^n + \alpha_n(y_n^a - x_n^a),\ \forall \alpha.$
Step 5 ***Convergence test.*** If convergence criterion is met, stop. The current solution, $\{x_a^n + 1\}$, is the set of equilibrium set flows. Otherwise, Set $n = n + 1$ and go to Step 1.

4.4 AUGMENTED FRANK-WOLFE ALGORITHM

The Frank-Wolfe Algorithm updates network flows using an arc perspective and lacks carrying path information. However, to examine whether a network could achieve an ϵ-UE state, path information is required for examining the definition of UE. That is, the arc-based FW algorithm can not determine whether a path belongs to the UE solutions. In fact, path information is hidden in all-or-nothing step. To overcome this weakness, Cho et al. [33] proposed the augmented Frank-Wolfe (AFW) algorithm. The design of AFW aims to eliminate the non-UE path flow. The AFW algorithm utilizes path information to switch a non-UE path flow and is described as follows:

First Stage

Step 1.1 Initialization. Set iteration counter $n = 0$ and $\forall a \in A$, use free flow travel cost C_a^0.

Step 1.2 Direction finding. Find each OD pair's shortest path p^{rs^n} and update arc flow $y_a^{(n)}$ via an AON assignment.

Step 1.3 Update arc and path information. Record these found shortest paths p^{rs^n} to the path set $\mathsf{P}_{\mathsf{rs}}^{(\mathsf{n})}$ and AON path flow $h_p^{rs^n}$, and set $f_n^{(a)} = y_n^{(a)}$.

Second Stage

Step 2.1 Update counter. Set iteration counter $n = n + 1$, $f_a^{(n)} = f_a^{(n-1)}$, and $h_p^{rs(n)} = h_p^{rs(n-1)}$.

Step 2.2 Direction finding. Set $C_a = C_A(f_a^{(n)})$, and find each OD pair's shortest path $p^{rs^n}$to update arc flow $y_a^{(n)}$ using an AON assignment.

Step 2.3 Step size determination. Find a step size; $\alpha^{(n)} = \underset{0\leq\alpha\leq1}{\arg\ \min}\ Z(f_a^{(n)} + \alpha.(y_a^{(n)} - f_a^{(n)}))$

Step 2.4 Update all path sets. If a shortest path $p^{rs^{(n)}}0 \notin P_{rs}^{(n-1)}$, update the path set $P_{rs}^{(n)} = P_{rs}^{(n-1)} \cup p^{rs^{(n)}}$, otherwise $P_{rs}^{(n)} = P_{rs}^{(n-1)}$.

Step 2.5 Update path flag. $\eta_p^{rs(n)} = 1$ if $p = p^{rs(n)}$, otherwise $\eta_p^{rs(n)} = 0$, $\forall p \in \{P_{rs}^{(n)} \backslash p^{rs(n)}\}$.

Step 2.6 Update arc flow. $f_a^{(n)} = f_a^{(n)} + \alpha^{(n)}.(y_a^{(n)} - f_a^{(n)})$.

Step 2.7 Update path flow. $h_p^{rs(n+1)} = h_p^{rs(n+1)} + \alpha^{(n)}.(q^{rs}\eta_p^{rs(n)} - h_p^{rs(n)})$

Step 2.8 Check whether a non-UE path flow exists. If a non-UE path flow exists, switch a non-EU path flow to the cheapest one; otherwise go to the next step.

Step 2.9 Convergence test. If $\epsilon - UE$ is met, then stop, otherwise go to Step 2.1.

The differences between AFW and FW algorithms are path information and examination of non-UE paths. Path information provides max-cost and min-cost paths. Furthermore, a max-cost path is used to examine whether a descent property is satisfied by switching a max-cost path flow to a cheapest cost one [33].

4.5 HAMILTONIAN PATH PROBLEM

The Hamiltonian Path Problem (HPP), a close cousin of Traveling Salesman Problem (TSP), can be posed as follows: Given a set $\{X_i : 1 \leq i \leq N\}$ of N nodes (cities) with starting and terminal nodes X_s and X_t, respectively and distances for each pair of nodes, what is the shortest path that starts at X_s and terminates at X_t and visits each node exactly once? Like TSP, HPP is also known to NP-hard.

It is interesting to note that the independent solutions to HPP are few. This is because the HPP can be solved using the solution to the TSP as follows:

1. The distance between X_s and X_t is set to an arbitrarily small value ($-\infty$), or,
2. A new node is added to the set of nodes with distances between is and X_s, as well as X_t are set to zero.

It is easy to see that since the path between X_s and X_t has to be included in the corresponding TSP, both of the above strategies will indeed yield a solution to the underlying HPP [34].

In this study, Hamiltonian Path Problem is solved via a genetic algorithm adapted to TSP named TSP_GA. TSP_GA is an available code that finds the TSP tour by using genetic algorithm. In this algorithm, solutions are obtained by only mutation operator. There are three different methods available in this algorithm. These are 2-opt method, node exchange method and node insertion method. After selecting the best solution found so far, these mutation methods are applied to this solution. Then, new solutions are compared again and the best solution is used in the next iteration. After finding a path, the longest link in the path is excluded to find the Hamiltonian path. Hamiltonian path is required because the alignment considered is not allowed to be circular, that is, there must be two separate start and terminus nodes.

4.6 AN ALGORITHM FOR MULTI-OBJECTIVE RTNDP

Standard network design techniques can not easily be applied to such contexts because the problem is typically of very large scale and involves non-linearity, as well as multiplicity of criteria [11]. To cope with such difficulties, we designed an algorithm that determines an alignment and stations on it as to minimize both construction cost and gas emission while preserving user equilibrium in the network. The main algorithm and two subroutines are shown in Figures 4.1 – 4.3.

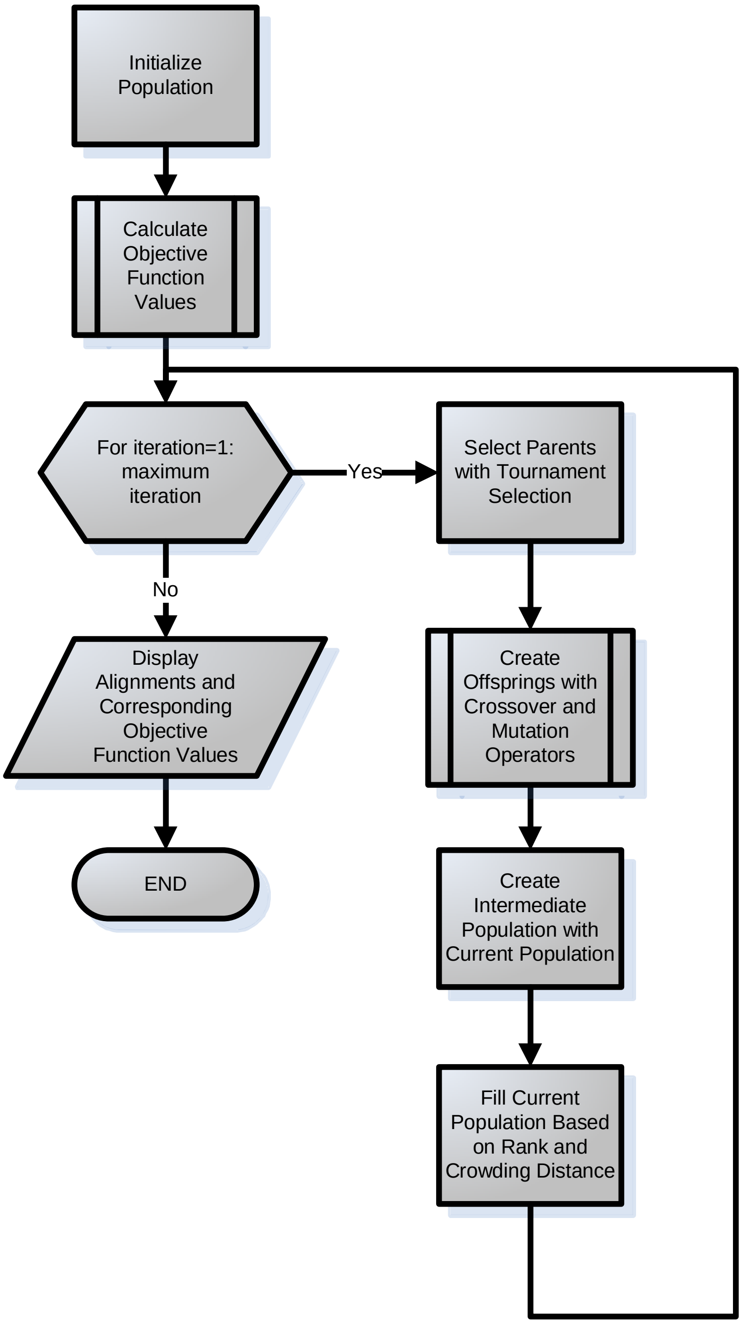


Figure 4.1 Main Algorithm

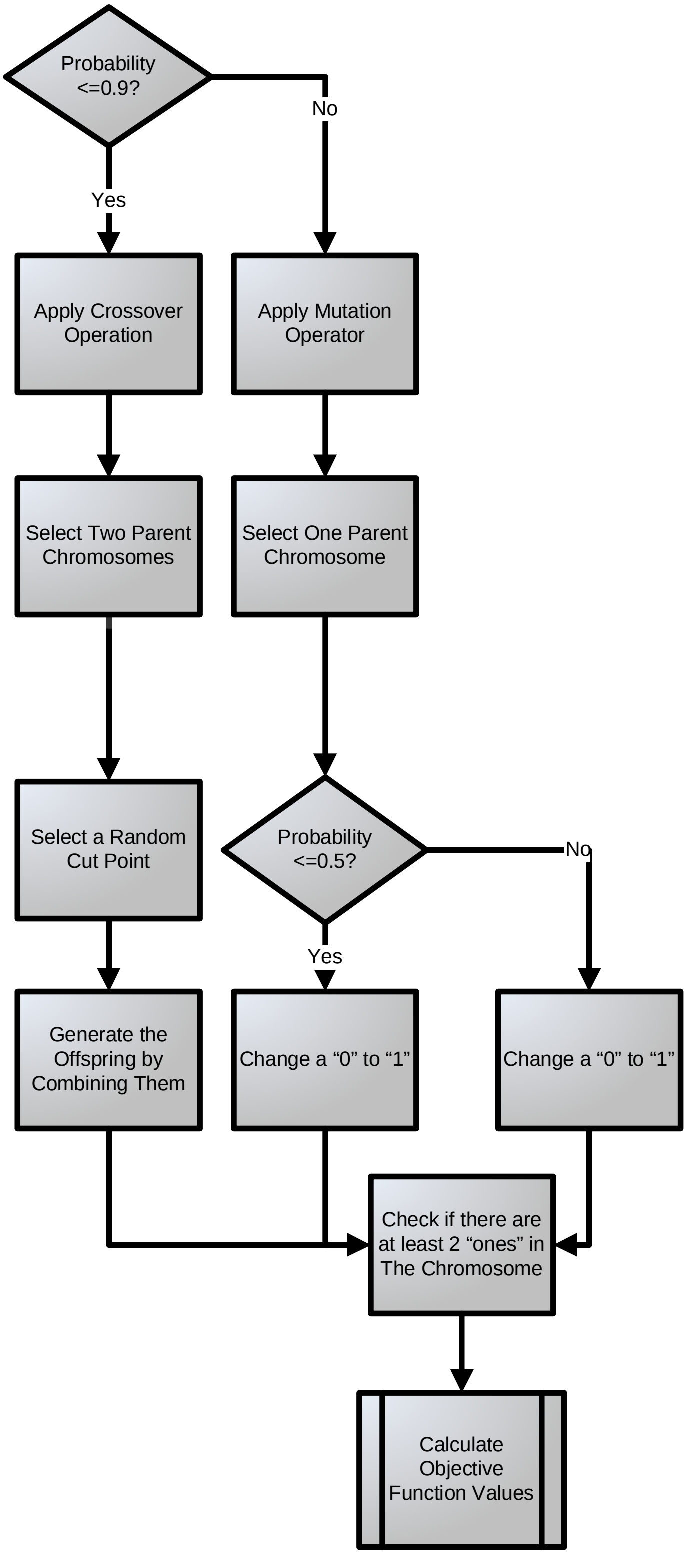


Figure 4.2 Creating Offsprings with Crossover and Mutation Operators

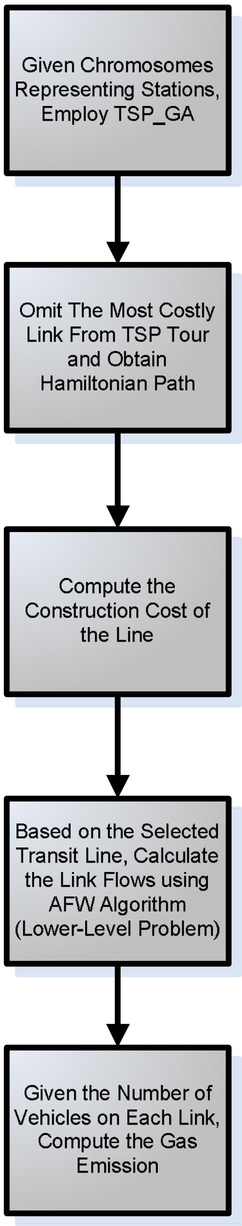


Figure 4.3 Calculating Objective Functions

The main algorithm is summarized below;

Step 1. Initialize population. Generate M (population size) chromosomes with N (number of nodes in the network) genes.

Step 2. Calculate objective function values.

Step 3. Set counter=1.

Step 4. Apply Tournament Selection Algorithms described in Section 4.1. Compare the solutions according to their objective function values. Randomly, select two individuals of which only one will be selected and added to the mating pool. Selection is performed based on two criteria. The first one is the front which the solution reside and the second one, if the solutions are at the same front, their crowding distances.

Step 5. Create offsprings with crossover and mutation operators. Assign a random number. If random number ≤ 0.9, go to Step 6, otherwise go to Step 7.

Step 6. Apply crossover operation. Select two parent chromosomes. Select a random cut point. Generate the offspring by combining the parent chromosomes. Go to Step 8.

Step 7. Apply mutation operation. Select one parent chromosome. If probability $\leq$ 0.5, change a “0” to “1” in the chromosome, otherwise change a “1” to “0” in the chromosome. Go to Step 8.

Step 8. Check if there are at least 2 “1”s in the chromosome to be able to construct a line.

Step 9. Calculate the objective function values as shown in Figure 4.3.

Step 10. Create intermediate population with current population and the offsprings.

Step 11. Apply non-dominated sorting algorithm described in Section 4.1.

Step 12. Fill the current population based on the front and crowding distance.

Step 13. counter=counter+1.

Step 14. If counter $\leq M$, go to step 4. Otherwise, stop.

5 NUMERICAL STUDY

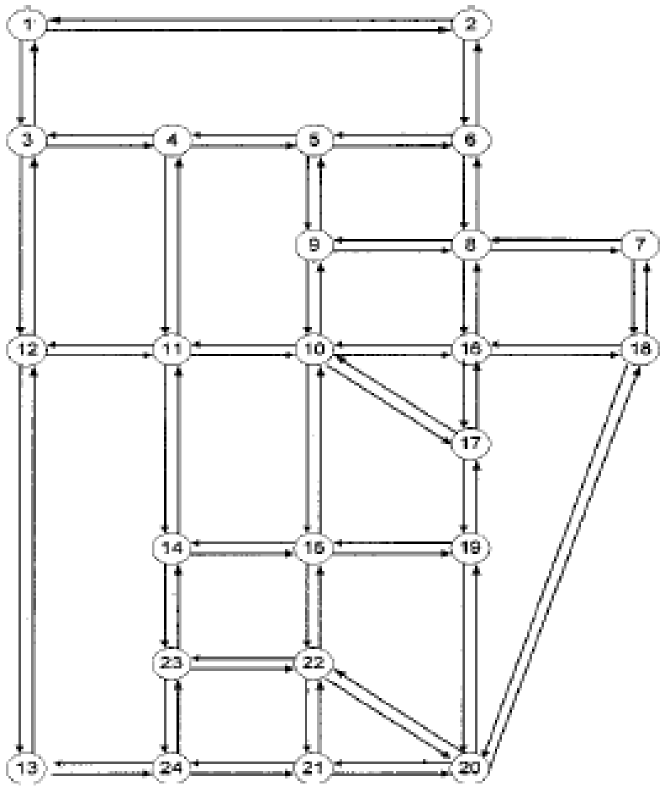


Figure 5.1 Sioux Falls Network

Our algorithm is tested on Sioux Falls Network which is first used by LeBlanc et al [35]. The network has 24 nodes, 76 links and 528 OD pairs. The parameters of each link are available in Table 5.2. This table shows the start and end nodes of each link, their link capacities and lengths, free flow times and other parameters that are used in (2.1) and the peak hour demands between network nodes.

Table 5.1 Link Parameters in Sioux Falls Network

Initial node	**Term node**	**Capacity** (c_{ij})	**Length** (l_{ij})	**Free Flow Time** (t^0_{ij})	α	**Power** β
1	2	25900.20	6	6	0.15	4
1	3	23403.47	4	4	0.15	4
2	1	25900.20	6	6	0.15	4
2	6	4958.18	5	5	0.15	4
3	1	23403.47	4	4	0.15	4
3	4	17110.52	4	4	0.15	4
3	12	23403.47	4	4	0.15	4
4	3	17110.52	4	4	0.15	4
4	5	17782.79	2	2	0.15	4
4	11	4908.82	6	6	0.15	4
5	4	17782.79	2	2	0.15	4
5	6	4947.99	4	4	0.15	4
5	9	10000.00	5	5	0.15	4
6	2	4958.18	5	5	0.15	4
6	5	4947.99	4	4	0.15	4
6	8	4898.58	2	2	0.15	4
7	8	7841.81	3	3	0.15	4
7	18	23403.47	2	2	0.15	4
8	6	4898.58	2	2	0.15	4
8	7	7841.81	3	3	0.15	4
8	9	5050.19	10	10	0.15	4
8	16	5045.82	5	5	0.15	4
9	5	10000.00	5	5	0.15	4
9	8	50501.93	10	10	0.15	4
9	10	13915.78	3	3	0.15	4
10	9	13915.78	3	3	0.15	4
10	11	10000.00	5	5	0.15	4
10	15	13512.00	6	6	0.15	4
10	16	4854.91	4	4	0.15	4
10	17	4993.51	8	8	0.15	4
11	4	4908.82	6	6	0.15	4
11	10	10000.00	5	5	0.15	4
11	12	4908.82	6	6	0.15	4
11	14	4876.50	4	4	0.15	4
12	3	23403.47	4	4	0.15	4
12	11	4908.82	6	6	0.15	4
12	13	25900.20	3	3	0.15	4
13	12	25900.20	3	3	0.15	4

Table 5.1 Link Parameters in Sioux Falls Network (continued)

Initial node	**Term node**	**Capacity (c_{ij})**	**Length (l_{ij})**	**Free Flow Time (t^0_{ij})**	α	**Power** β
13	24	5091.25	4	4	0.15	4
14	11	4876.50	4	4	0.15	4
14	15	5127.52	5	5	0.15	4
14	23	4924.79	4	4	0.15	4
15	10	13512.00	6	6	0.15	4
15	14	5127.52	5	5	0.15	4
15	19	14564.75	3	3	0.15	4
15	22	9599.18	3	3	0.15	4
16	8	5045.82	5	5	0.15	4
16	10	4854.91	4	4	0.15	4
16	17	5229.91	2	2	0.15	4
16	18	19679.89	3	3	0.15	4
17	10	4993.51	8	8	0.15	4
17	16	5229.91	2	2	0.15	4
17	19	4823.95	2	2	0.15	4
18	7	23403.47	2	2	0.15	4
18	16	19679.89	3	3	0.15	4
18	20	23403.47	4	4	0.15	4
19	15	14564.75	3	3	0.15	4
19	17	4823.95	2	2	0.15	4
19	20	5002.60	4	4	0.15	4
20	18	23403.47	4	4	0.15	4
20	19	5002.60	4	4	0.15	4
20	21	5059.91	6	6	0.15	4
20	22	5075.69	5	5	0.15	4
21	20	5059.91	6	6	0.15	4
21	22	5229.91	2	2	0.15	4
21	24	4885.35	3	3	0.15	4
22	15	9599.18	3	3	0.15	4
22	20	5075.69	5	5	0.15	4
22	21	5229.91	2	2	0.15	4
22	23	5000.00	4	4	0.15	4
23	14	4924.79	4	4	0.15	4
23	22	5000.00	4	4	0.15	4
23	24	5078.50	2	2	0.15	4
24	13	5091.25	4	4	0.15	4
24	21	4885.35	3	3	0.15	4
24	23	5078.50	2	2	0.15	4

Table 5.2 OD Demands

ORIGIN \ DESTINATION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	100	100	500	200	300	500	800	500	1300	500	200	500	300	500	500	400	100	300	300	100	400	300	100
2	100	0	100	200	100	400	200	400	200	600	200	100	300	100	100	400	200	0	100	100	0	100	0	0
3	100	100	0	200	100	300	100	200	100	300	300	200	100	100	100	200	100	0	0	0	0	100	100	0
4	500	200	200	0	500	400	400	700	700	1200	1400	600	600	500	500	800	500	100	200	300	200	400	500	200
5	200	100	100	500	0	200	200	500	800	1000	500	200	200	100	200	500	200	0	100	100	100	200	100	0
6	300	400	300	400	200	0	400	800	400	800	400	200	200	100	200	900	500	100	200	300	100	200	100	100
7	500	200	100	400	200	400	0	100	600	1900	500	700	400	200	500	1400	1000	200	400	500	200	500	200	100
8	800	400	200	700	500	800	1000	0	800	1600	800	600	600	400	600	2200	1400	300	700	900	400	500	300	200
9	500	200	100	700	800	400	600	800	0	2800	1400	600	600	600	900	1400	900	200	400	600	300	700	500	200
10	1300	600	300	1200	1000	800	1900	1600	2800	0	4000	2000	1900	2100	4000	4400	3900	700	1800	2500	1200	2600	1800	800
11	500	200	300	1500	500	400	500	800	1400	3900	0	1400	1000	1600	1400	1400	1000	100	400	600	400	1100	1300	600
12	200	100	200	600	200	200	700	600	600	2000	1400	0	1300	700	700	700	600	200	300	400	300	700	700	500
13	500	300	100	600	200	200	400	600	600	1900	1000	1300	0	600	700	600	500	100	300	600	600	1300	800	800
14	300	100	100	500	100	100	200	400	600	2100	1600	700	600	0	1300	700	700	100	300	500	400	1200	1100	400
15	500	100	100	500	200	200	500	600	1000	4000	1400	700	700	1300	0	1200	1500	200	800	1100	800	2600	1000	400
16	500	400	200	800	500	900	1400	2200	1400	4400	1400	700	600	700	1200	0	2800	500	1300	1600	600	1200	500	300
17	400	200	100	500	200	500	1000	1400	900	3900	1000	600	500	700	1500	2800	0	600	1700	1700	600	1700	600	300
18	100	0	0	100	0	100	200	300	200	700	200	200	100	100	200	500	600	0	300	400	100	300	100	0
19	300	100	0	200	100	200	400	700	400	1800	400	300	300	300	800	1300	1700	300	0	1200	400	1200	300	100
20	300	100	0	300	100	300	500	900	600	2500	600	500	600	500	1100	1600	1700	400	1200	0	1200	2400	700	400
21	100	0	0	200	100	100	200	400	300	1200	400	300	600	400	800	600	600	100	400	1200	0	1800	700	500
22	400	100	100	400	200	200	500	500	700	2600	1100	700	1300	1200	2600	1200	1700	300	1200	2400	1800	0	2100	1100
23	300	0	100	500	100	100	200	300	500	1800	1300	700	800	1100	1000	500	600	100	300	700	700	2100	0	700
24	100	0	0	200	0	100	100	200	200	800	600	500	700	400	400	300	300	0	100	400	500	1100	700	0

Table 5.3 Construction Costs of Links Between Nodes

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.00	270.00	70.00	106.30	183.85	278.,93	392.17	299.67	214.01	254.95	206.16	190.00
2	270.00	0.00	278.93	202.48	122.07	70.00	164.01	130.00	164.01	214.71	268.70	330.15
3	70.00	278.93	0.00	80.00	170.00	270.00	374.83	276.59	180.28	208.09	144.22	120.00
4	106.30	202.48	80.00	0.00	90.00	190.00	296.14	199.25	108.17	150.00	120.00	144.22
5	183.85	122.07	170.00	90.00	0.00	100.00	208.81	116.62	60.00	120.00	150.00	208.09
6	278.93	70.00	270.00	190.00	100.00	0.00	116.62	60.00	116.62	156.21	224.72	295.47
7	392.17	164.01	374.83	296.14	208.81	116.62	0.00	100.00	200.00	208.81	296.14	374.83
8	299.67	130.00	276.59	199.25	116.62	60.00	100.00	0.00	100.00	116.62	199.25	276.59
9	214.01	164.01	180.28	108.17	60.00	116.62	200.00	100.00	0.00	60.00	108.17	180.28
10	254.95	214.71	208.09	150.00	120.00	156.21	208.81	116.62	60.00	0.00	90.00	170.00
11	206.16	268.70	144.22	120.00	150.00	224.72	296.14	199.25	108.17	90.00	0.00	80.00
12	190.00	330.15	120.00	144.22	208.09	295.47	374.83	276.59	180.28	170.00	80.00	0.00
13	460.00	533.39	390.00	398.12	425.44	474.34	495.78	426.38	371.21	319.06	281.60	270.00
14	329.85	372.16	262.49	250.00	265.71	314.01	346.70	268.70	210.24	158.11	130.00	152.64
15	362.35	335.26	302.32	265.71	250.00	269.26	275.86	214.71	190.00	130.00	158.11	214.01
16	330.15	190.00	295.47	224.72	156.21	120.00	116.62	60.00	116.62	100.00	190.00	270.00
17	367.97	250.00	324.50	261.73	205.91	180.00	156.21	120.00	156.21	116.62	199.25	276.59
18	415.93	214.71	388.97	313.85	233.24	156.21	60.00	116.62	208.81	200.00	290.00	370.00
19	418.69	320.00	367.97	314.01	269.26	250.00	214.71	190.00	214.71	164.01	230.22	299.67
20	533.39	460.00	474.34	433.82	402.62	390.00	344.82	330.00	344.82	287.92	330.15	381.84
21	490.41	470.74	425.44	400.25	390.00	402.62	385.88	344.82	330.00	270.00	284.61	319.06
22	416.29	392.94	353.55	322.80	310.00	325.73	320.16	269.26	250.00	190.00	210.24	254.95
23	388.33	424.85	320.16	310.00	322.80	363.59	382.88	314.01	265.71	210.24	190.00	206.16
24	466.90	497.69	398.12	390.00	400.25	433.82	439.32	380.79	342.05	284.61	270.00	281.60

Table 5.3 Construction Costs of Links Between Nodes (continued)

	13	14	15	16	17	18	19	20	21	22	23	24
1	460.00	329.85	362.35	330.15	367.97	415.93	418.69	533.39	490.41	416.29	388.33	466.90
2	533.39	372.16	335.26	190.00	250.00	214.71	320.00	460.00	470.74	392.94	424.85	497.69
3	390.00	262.49	302.32	295.47	324.50	388.97	367.97	474.34	425.44	353.55	320.16	398.12
4	398.12	250.00	265.71	224.72	261.73	313.85	314.01	433.82	400.25	322.80	310.00	390.00
5	425.44	265.71	250.00	156.21	205.91	233.24	269.26	402.62	390.00	310.00	322.80	400.25
6	474.34	314.01	269.26	120.00	180.00	156.21	250.00	390.00	402.62	325.73	363.59	433.82
7	495.78	346.70	275.86	116.62	156.21	60.00	214.71	344.82	385.88	320.16	382.88	439.32
8	426.38	268.70	214.71	60.00	120.00	116.62	190.00	330.00	344.82	269.26	314.01	380.79
9	371.21	210.24	190.00	116.62	156.21	208.81	214.71	344.82	330.00	250.00	265.71	342.05
10	319.06	158.11	130.00	100.00	116.62	200.00	164.01	287.92	270.00	190.00	210.24	284.61
11	281.60	130.00	158.11	190.00	199.25	290.00	230.22	330.15	284.61	210.24	190.00	270.00
12	270.00	152.64	214.01	270.00	276.59	370.00	299.67	381.84	319.06	254.95	206.16	281.60
13	0.00	161.25	220.23	381.84	342.05	458.04	304.14	270.00	170.00	187.88	113.14	80.00
14	161.25	0.00	90.00	230.22	202.48	317.81	190.00	236.01	166.43	108.17	60.00	140.00
15	220.23	90.00	0.00	164.01	122.07	238.54	100.00	172.05	140.00	60.00	108.17	166.43
16	381.84	230.22	164.01	0.00	60.00	100.00	130.00	270.00	287.92	214.71	268.70	330.15
17	342.05	202.48	122.07	60.00	0.00	116.62	70.00	210.00	232.59	164.01	230.22	283.20
18	458.04	317.81	238.54	100.00	116.62	0.00	164.01	287.92	336.01	275.86	346.70	396.23
19	304.14	190.00	100.00	130.00	70.00	164.01	0.00	140.00	172.05	116.62	199.25	236.01
20	270.00	236.01	172.05	270.00	210.00	287.92	140.00	0.00	100.00	128.06	206.16	190.00
21	170.00	166.43	140.00	287.92	232.59	336.01	172.05	100.00	0.00	80.00	120.42	90.00
22	187.88	108.17	60.00	214.71	164.01	275.86	116.62	128.06	80.00	0.00	90.00	120.42
23	113.14	60.00	108.17	268.70	230.22	346.70	199.25	206.16	120.42	90.00	0.00	80.00
24	80.00	140.00	166.43	330.15	283.20	396.23	236.01	190.00	90.00	120.42	80.00	0.00

Table 5.4 Travel Times of Possible Transit Links between Nodes

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	0.00	10.56	7.04	14.08	17.59	19.35	28.15	22.87	26.39	31.67	24.63	14.08
2	10.56	0.00	17.59	19.35	15.83	8.80	17.59	12.32	24.63	28.15	29.91	24.63
3	7.04	17.59	0.00	7.04	10.56	17.59	26.39	21.11	19.35	24.63	17.59	7.04
4	14.08	19.35	7.04	0.00	3.52	10.56	19.35	14.08	12.32	17.59	10.56	14.08
5	17.59	15.83	10.56	3.52	0.00	7.04	15.83	10.56	8.80	14.08	14.08	17.59
6	19.35	8.80	17.59	10.56	7.04	0.00	8.80	3.52	15.83	19.35	21.11	24.63
7	28.15	17.59	26.39	19.35	15.83	8.80	0.00	5.28	21.11	15.83	24.63	33.43
8	22.87	12.32	21.11	14.08	10.56	3.52	5.28	0.00	17.59	15.83	24.63	28.15
9	26.39	24.63	19.35	12.32	8.80	15.83	21.11	17.59	0.00	5.28	14.08	24.63
10	31.67	28.15	24.63	17.59	14.08	19.35	15.83	15.83	5.28	0.00	8.80	19.35
11	24.63	29.91	17.59	10.56	14.08	21.11	24.63	24.63	14.08	8.80	0.00	10.56
12	14.08	24.63	7.04	14.08	17.59	24.63	33.43	28.15	24.63	19.35	10.56	0.00
13	19.35	29.91	12.32	19.35	22.87	29.91	33.43	33.43	29.91	24.63	15.83	5.28
14	31.67	36.95	24.63	17.59	21.11	28.15	29.91	29.91	21.11	15.83	7.04	17.59
15	40.47	33.43	33.43	26.39	24.63	24.63	21.11	21.11	15.83	10.56	15.83	26.39
16	31.67	21.11	29.91	22.87	19.35	12.32	8.80	8.80	12.32	7.04	15.83	26.39
17	35.19	24.63	33.43	26.39	22.87	15.83	12.32	12.32	15.83	10.56	19.35	29.91
18	31.67	21.11	29.91	22.87	19.35	12.32	3.52	8.80	17.59	12.32	21.11	31.67
19	38.71	28.15	36.95	29.91	26.39	19.35	15.83	15.83	19.35	14.08	21.11	31.67
20	38.71	28.15	35.19	29.91	26.39	19.35	10.56	15.83	24.63	19.35	28.15	28.15
21	31.67	38.71	24.63	31.67	33.43	29.91	21.11	26.39	24.63	19.35	22.87	17.59
22	35.19	36.95	28.15	31.67	29.91	28.15	19.35	24.63	21.11	15.83	21.11	21.11
23	29.91	40.47	22.87	24.63	28.15	35.19	26.39	31.67	28.15	22.87	14.08	15.83
24	26.39	36.95	19.35	26.39	29.91	35.19	26.39	31.67	29.91	24.63	17.59	12.32

Table 5.4 Travel Times of Possible Links between Nodes (continued)

	13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
1	19.35	31.67	40.47	31.67	35.19	31.67	38.71	38.71	31.67	35.19	29.91	26.39
2	29.91	36.95	33.43	21.11	24.63	21.11	28.15	28.15	38.71	36.95	40.47	36.95
3	12.32	24.63	33.43	29.91	33.43	29.91	36.95	35.19	24.63	28.15	22.87	19.35
4	19.35	17.59	26.39	22.87	26.39	22.87	29.91	29.91	31.67	31.67	24.63	26.39
5	22.87	21.11	24.63	19.35	22.87	19.35	26.39	26.39	33.43	29.91	28.15	29.91
6	29.91	28.15	24.63	12.32	15.83	12.32	19.35	19.35	29.91	28.15	35.19	35.19
7	33.43	29.91	21.11	8.80	12.32	3.52	15.83	10.56	21.11	19.35	26.39	26.39
8	33.43	29.91	21.11	8.80	12.32	8.80	15.83	15.83	26.39	24.63	31.67	31.67
9	29.91	21.11	15.83	12.32	15.83	17.59	19.35	24.63	24.63	21.11	28.15	29.91
10	24.63	15.83	10.56	7.04	10.56	12.32	14.08	19.35	19.35	15.83	22.87	24.63
11	15.83	7.04	15.83	15.83	19.35	21.11	21.11	28.15	22.87	21.11	14.08	17.59
12	5.28	17.59	26.39	26.39	29.91	31.67	31.67	28.15	17.59	21.11	15.83	12.32
13	0.00	17.59	21.11	31.67	29.91	29.91	26.39	22.87	12.32	15.83	10.56	7.04
14	17.59	0.00	8.80	21.11	17.59	26.39	14.08	21.11	15.83	14.08	7.04	10.56
15	21.11	8.80	0.00	12.32	8.80	17.59	5.28	12.32	8.80	5.28	12.32	14.08
16	31.67	21.11	12.32	0.00	3.52	5.28	7.04	12.32	21.11	17.59	24.63	26.39
17	29.91	17.59	8.80	3.52	0.00	8.80	3.52	10.56	17.59	14.08	21.11	22.87
18	29.91	26.39	17.59	5.28	8.80	0.00	12.32	7.04	17.59	15.83	22.87	22.87
19	26.39	14.08	5.28	7.04	3.52	12.32	0.00	7.04	14.08	10.56	17.59	19.35
20	22.87	21.11	12.32	12.32	10.56	7.04	7.04	0.00	10.56	8.80	15.83	15.83
21	12.32	15.83	8.80	21.11	17.59	17.59	14.08	10.56	0.00	3.52	8.80	5.28
22	15.83	14.08	5.28	17.59	14.08	15.83	10.56	8.80	3.52	0.00	7.04	8.80
23	10.56	7.04	12.32	24.63	21.11	22.87	17.59	15.83	8.80	7.04	0.00	3.52
24	7.04	10.56	14.08	26.39	22.87	22.87	19.35	15.83	5.28	8.80	3.52	0.00

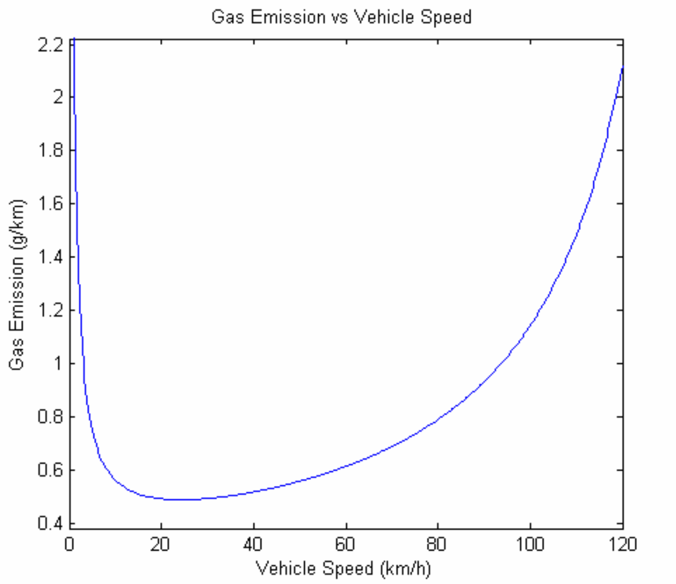


Figure 5.2 Amount of Gas Emission Corresponding to Vehicle Speed

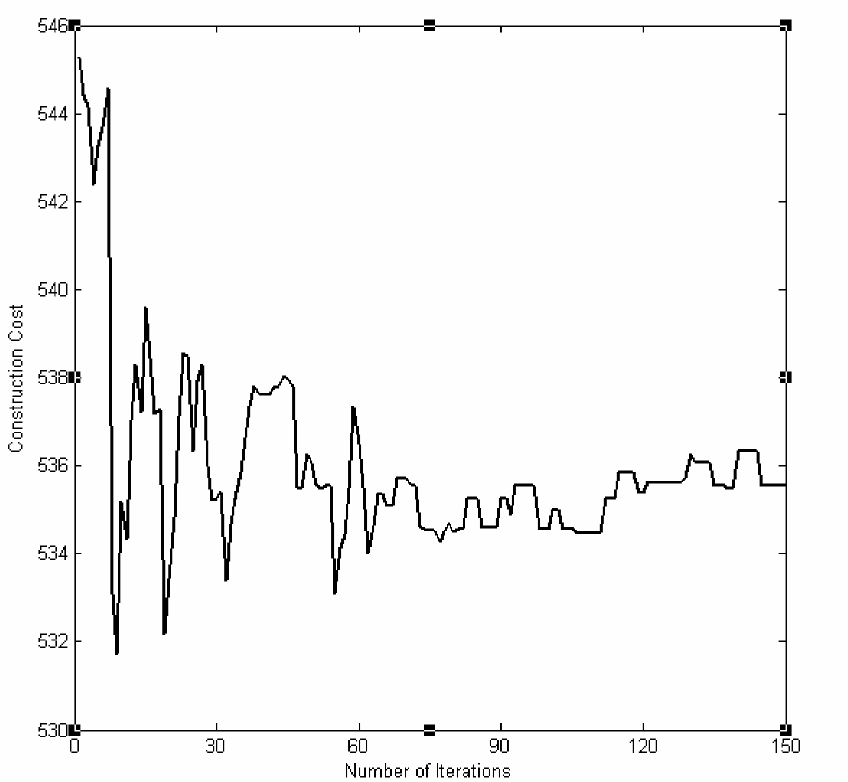


Figure 5.3 Change in Transit Network Construction Cost in Each Iteration

Figure 5.2 shows the change in gas emission of one vehicle as its speed increases. It can be easily seen that amount of gas emission is decreasing in the first portion of the graph and then it increases regularly as the speed of the vehicle increases. That means gas emission in a link is at the top level when traffic is congested. However, when vehicle travels with high speed, it exposes, again, a high level of harmful gases. The model calibrates the number of vehicles in each link, thereby their speeds, so as to minimize the gas emission and the construction cost of the network.

Figures 5.3 and 5.4 explain why number of iterations is fixed at 150 in Genetic Algorithm. These figures show the variation in the objective function means, construction cost and gas emission respectively, calculated over all population. Obviously, after approximately 100 iterations, the population almost converges. In fact, up to this point, changes in both objective functions become quite small. But for the sake of the quality of the results, the number of iterations is extended to 150. In addition, to provide diversity and obtain better solutions using strong ones, the probability of crossover and mutation in NSGA-II algorithm is determined as 0.9 and 0.1, respectively.

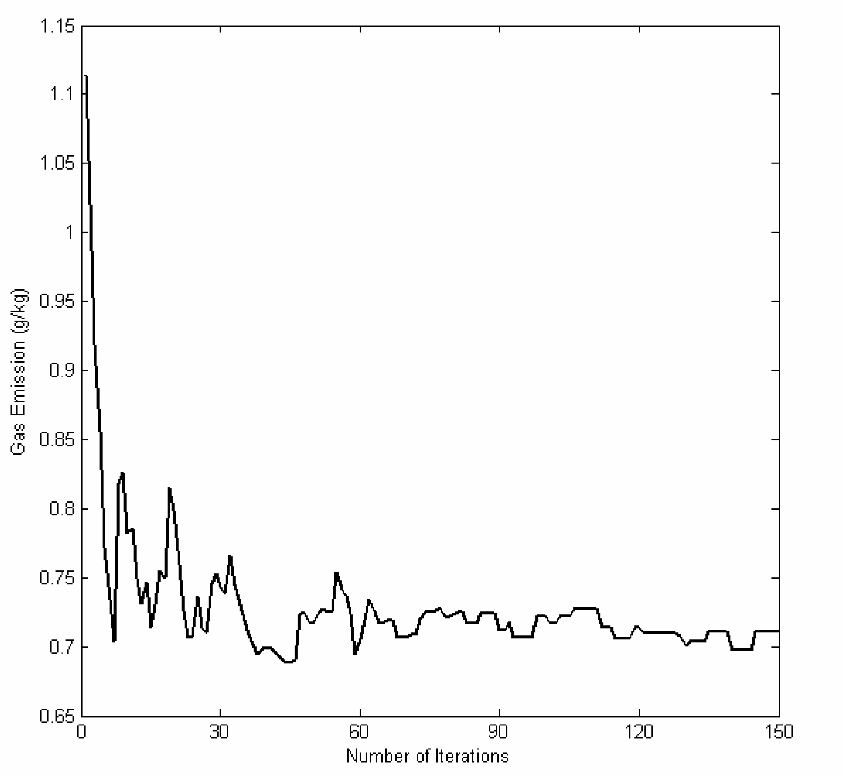


Figure 5.4 Change in Gas Emission in Each Iteration

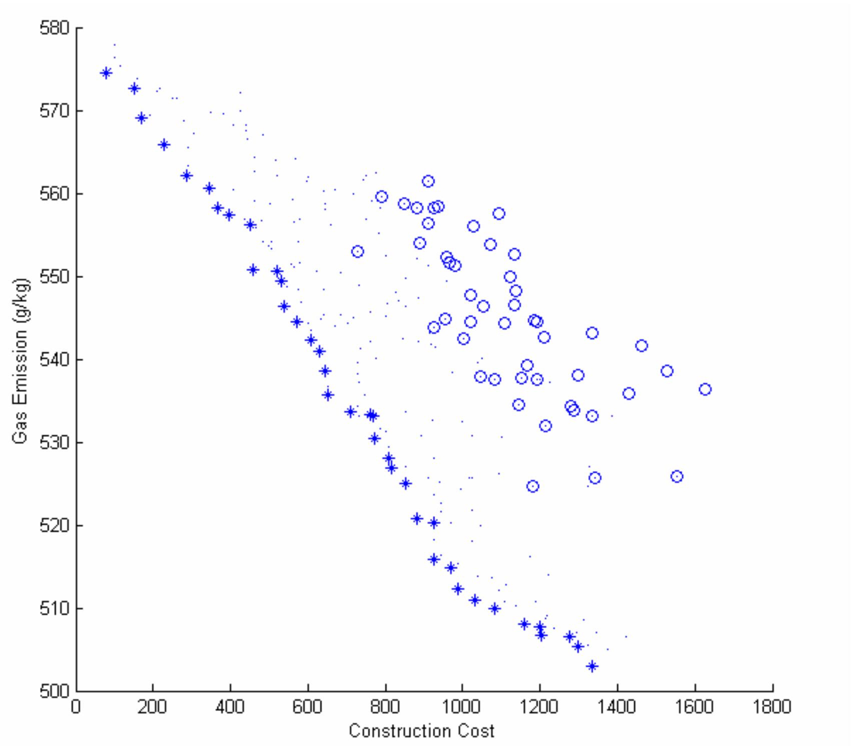


Figure 5.5 Pareto Front of Our Algorithm

Figure 5.5 shows the Pareto Front of our results. The solutions represented by circles are the initial solutions that are selected randomly to begin the algorithm. In each iteration, solutions get closer to Pareto Front and this behavior is pictured with dots between circles and stars. It can be also seen that diversity is provided as the number of iterations increase besides obtaining non-dominated solutions. Final solutions also seem to be satisfactory because many solutions are found in a wide range.

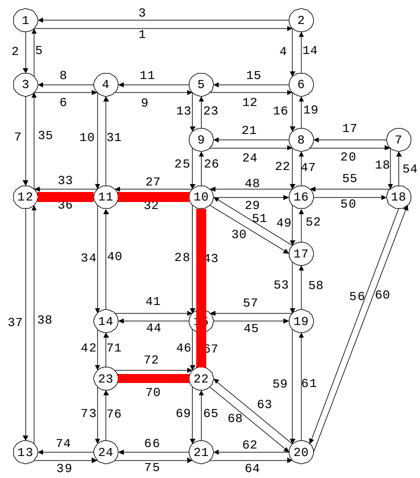
Solutions in the non-dominated set and corresponding values of each objective function are shown in the Table 5.3. From these data, reduction in the gas emission for different number of station can be observed. The original gas emission amount in the network when transit line is not available is 577,469 (g/km). The amount of reduction of each solution is also available in the Table 5.3. It can be seen that as the number of stations increases, the gas emission in the network reduces. If faster trains are used in the alignment, it may be reduced further since more people will tend to use transit line.

Table 5.5 Non-Dominated Solutions and Corresponding Objective Function Values

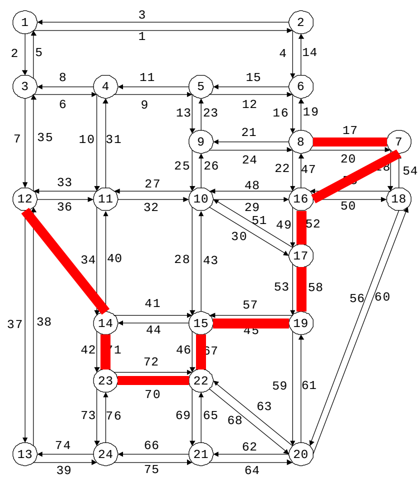
Construction Cost	**Gas Emission (g/km)**	**Reduction in Gas Emission**	**# of Stations**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
80	574.5	0.51%	2	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
150	572.6	0.84%	3	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
170	569.1	1.45%	3	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
230	565.8	2.02%	4	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
286.6	562.1	2.66%	4	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
346.6	560.6	2.92%	5	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
366.6	558.2	3.34%	6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	0
396.6	557.4	3.48%	6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0
450	556.2	3.68%	5	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0
456.6	550.7	4.64%	7	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	0	1	0	0	1	1	0
522.1	550.5	4.67%	7	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	0	1	1	0
532.6	549.5	4.84%	7	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	1	0	1	0	0	0	0	0
540	546.3	5.40%	8	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	1	0	0	1	1	0
570	544.5	5.71%	8	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0
609.3	542.3	6.09%	8	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	0
630	540.9	6.33%	9	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	1	0	1	0	0	1	1	0
646.2	538.5	6.75%	8	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	1	0	0	1	1	0
652.6	535.7	7.23%	9	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	1	0	1	0	0	1	1	0
710	533.6	7.60%	10	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	1	0	0	1	1	0

Table 5.3 Non-dominated solutions and corresponding objective function values (continued)

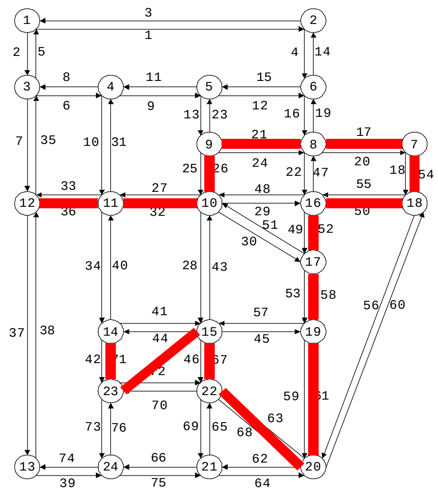
Construction Cost	Gas Emission (g/km)	Reduction in Gas Emission	# of Stations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
763.1	533.3	7.65%	10	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	1	0	1	0	1	1	1	0
769.3	533.1	7.68%	10	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	1	0	0	1	1	0
773.2	530.5	8.13%	10	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	1	1	0	1	0	0	1	1	0
809.3	528.1	8.55%	10	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	1	0	1	0	0	1	1	0
816.6	526.9	8.76%	11	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	1	1	1	1	0	0	1	1	0
852.6	525	9.09%	11	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	1	1	1	0	0	1	1	0
883.2	520.8	9.81%	11	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	1	1	0	1	0	0	1	1	0
925.9	520.2	9.92%	11	0	0	0	0	0	0	1	1	0	1	0	1	0	1	1	1	1	0	1	0	0	1	1	0
926.6	515.8	10.68%	12	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	1	1	1	1	0	0	1	1	0
970	514.7	10.87%	13	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	0
986.6	512.4	11.27%	13	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	1	1	1	1	0	0	1	1	0
1030	510.9	11.53%	14	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	0
10831	509.9	11.70%	14	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1	0
11598	508.1	12.01%	15	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1	0
11981	507.7	12.08%	15	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	0
12031	506.7	12.26%	16	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1	0
1274.7	506.6	12.27%	16	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	0
1297.3	505.3	12.50%	16	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	0
1333.8	503	12.90%	16	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0



(a)



(b)



(c)

Figure 5.6 Examples of Transit Lines Constructed by the Algorithm

Population size to be used in NSGA-II is taken as 50. The reason why 38 solutions exist in Table 5.3 is that some solutions are repeated in the population. Figure 5.6 shows 3 examples of transit line constructed by the algorithm. Figure 5.6 (a) is a transit line with 5 stations while 5.6(b) and 5.6(c) are transit lines with 10 and 15 stations respectively. As expected, as the number of stations in an alignment because more OD pairs will be connected by this line. In these examples, the construction costs are 450, 809.3 and 1198.1 respectively. Consequently, their gas emissions are inversely proportional with their costs. Emission rates released to the air are 556.2, 528.1 and 507.7 (g/km) for each solution.

6 CONCLUSION

In this study, a multi-objective bi-level model is proposed because there are two contradicting objectives in the problem. The optimum solution for minimizing gas emission is probably constructing a line that passes through all nodes in the network so it is possible to travel between each OD pair by using rapid transit line. In contrary, to minimize investment cost of the alignment, no line, as a result no station should be constructed. To find a compromising solution between these two objectives, a multi-objective approach is introduced. The result for this problem is a Pareto-Chart that shows different solutions with different objective function values that allows decision makers to select one according to their priorities. Because the problem is a Stackelberg game, to take user behavior into account, the model is formed as a bi-level model and the municipality which is the leader of this game, makes changes to affect users' decisions.

Minimizing gas emission is introduced into RTNDP for the first time, although there are many studies on this topic. Most of these studies use maximizing trip coverage, maximizing population covered while some of them also consider minimizing the construction cost in a normalized equation with other objectives. The originality of this study mainly comes from that environmental impacts are taken into account for the first time. Air pollution is a serious problem, especially for crowded cities and since gas emission from vehicles is a major source for air pollution, rapid transit lines are considered as a solution for air pollution because of its high potential to reduce traffic congestion.

RTNDP is a fruitful subject that can be studied further. This study primarily focuses on the deterministic aspect of the problem, in other words, all users in the network have perfect information about the travel times in each alternative they can choose. A stochastic model that enables users choose their routes and travel modes according to their perceived travel times rather than the actual travel times may be developed. The perceived travel time can be looked upon as a random variable distributed across the users, that is, each motorist may perceive a different travel time over the same link and this definition characterizes the stochastic-user-equilibrium (SUE) condition.

A dynamic model that analyzes the network over a specific period is another direction for further research. Our study generates solutions by including demands and corresponding travel times in peak hours. Instead of making computations according to the maximum demands possible, the network may be analyzed for a period so that a thorough analysis of the user behavior may be obtained.

Furthermore, in this study park-and-ride and kiss-and-ride alternatives do not exist; a user traveling between an OD pair is not allowed to travel from his/her starting node to the nearest station by his/her car and then continue the remaining part by transit line. Rapid transit lines can be chosen if and only if there exist stations on both origin and destination nodes of the users. This assumption is restrictive because it does not consider potential users of the transit line. It may be relaxed to obtain more realistic results because these are widely used methods and they are also encouraged by municipalities to increase the effectiveness of the lines. Parking zones are available, especially near busy stations, thus it is assured that users starting their travels from nodes that do not have stations can also use the line.

As a last point, it is also possible to construct multiple lines instead of one. This extension may lead a more flexible solution and more reduction in emission minimizing objective while keeping cost minimizing objective constant may be obtained or vice versa.

REFERENCES

[1] Sheffi, Y., *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, Prentice Hall Inc., (1985).

[2] Schöbel, A., *Optimization in Public Transportation*, Springer, (2006).

[3] Karoonsoontawong, A., *Robustness Approach to the Integrated Network Design Problem, Signal Optimization and Dynamic Traffic Assignment Problem*, Ph. D. Thesis, University of Texas, Austin, (2006).

[4] Marin, A., "An Extension to Rapid Transit Network Design Problem", *TOP*, 15, 231-241, (2007).

[5] Magnanti, T. L., Wong, R. T., "Network Design and Transportation Planning: Models and Algorithms", *Transportation Science*, 18 (1), 1-55, (1984).

[6] Bruno, G., Ghiani, G., Improta, G., "A Multi-modal Approach to the Location of a Rapid Transit Line", *European Journal of Operational Research*, 104, 321-332, (1998).

[7] Karlaftis, M. G., "A DEA Approach for Evaluating the Efficiency and Effectiveness of Urban Transit Systems", *European Journal of Operations Research*, 152, 354-364, (2004).

[8] Laporte, G., Mesa, J.A., Ortega, F.A., Sevillano, I., "Maximizing Trip Coverage in the Location of a Single Rapid Ttransit Alignment", *Annals of Operations Research*, 136, 49–63, (2005).

[9] Hamacher, H. W., Liebers, A., Schöbel, A., Wagner, D., Wagner, F., “Locating New Stops in a Railway Network”, *Electronic Notes in Theoretical Computer Science*, 50 (1), 1-10, (2001).

[10] Laporte, G., Mesa, J.A., Ortega, F.A., “Locating Stations on Rapid Transit Lines”, *Computers & Operations Research*, 29, 741–759, (2002).

[11] Bruno, G., Gendreau, M, Laporte, G, “A heuristic for the location of a rapid transit line”, *Computers & Operations Research*, 29, 1–12, (2002).

[12] Mesa, J.A., Ortega, F. A., “Park and Ride Station Catchment Areas in Metropolitan Rapid Transit Systems”, *Mathematical Models on Optimization in Transportation Systems*. In M. Pursula and J. Niittyemäki (eds), Kluwer, Dordrecht, 81-93, (2001) .

[13] Laporte, G., Marin, A., Mesa, J. A., Ortega, F. A., “An Integrated Methodology for the Rapid Transit Network Design Problem”, In F. Garaets, L. Kroon , A. Schoebel, D. Wagner, & CH. Zaroliagis (EDS), *Lecture Notes in Computer Science*, 4359, Algorithmic Methods for Railway Optimization , Berlin, Springer, (2007).

[14] Marin, A., Jaramillo, P., “Urban Rapid Transit Network Design: Accelerated Benders Decomposition”, *Annals of Operations Research*, 169, 35-53, (2009).

[15] Escudero, L. F., Munoz, S., “An Approach for Solving a Modification of the Extended Rapid Transit Network Design Problem”, *TOP*, 17, 320-334, (2009).

[16] Ntziachristos, N., Samaras, Z., “Methodology for the Calculation of Exhaust Emissions”, http://lat.eng.auth.gr/copert/, (2009).

[17] Bard, J. F., *Practical Bi-level Optimization, Algorithms and Applications*, Kluwer Academic Publishers, (1998).

[18] Colson, B., Marcotte, P., Savard, G., “An Overview of Bi-level Optimization”, *Annals of Operations Research*, 153, 235-256, (2007).

[19] Dempe, S., “Annotated Bibliography on Bi-level Programming and Mathematical Programs with Equilibrium Constraints”, *Optimization*, Vol. 52, No.3, 333-359, (2003).

[20] Jeroslaw, R. G., “The Polynomial Hierarchy and a Simple Model for Competitive Analysis”, *Mathematical programming*, 32, 146-164, (1985).

[21] Yin, Y., “Genetic-Algorithms-Based Approach for Bi-Level Programming Models”, *Journal of Transportation Engineering*, 126, No.2, 115-120, (2000)

[22] Deb K., *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons Ltd, (2002).

[23] Deb, K., Pratap A., Agarwal S., Meyarivan T., “A Fast and Elitist Multi objective Genetic Algorithm: NSGA-II”, *IEEE Transactions on Evolutionary Computation*, 6, 182-197, (2002).

[24] Schaffer, J. D., *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms*, Ph. D. Thesis, Nashville, TN: Vanderbilt University, USA, (1984).

[25] Schaffer, J. D., “Multiple Objective Optimization with Vector Evaluated Genetic Algorithms”, *In Proceedings of the First International Conference on Genetic Algorithms*, 93-100, (1985).

[26] Goldberg, D. E., *Genetic Algorithms for Search, Optimization and Machine Learning*, Reading, MA, Addison Wesley, (1989).

[27] Srinivas, N., Deb K., “Multi-Objective Function Optimization Using Non-Dominated Sorting Genetic Algorithms”, *Evolutionary Computation Journal*, 2 (3), 221-248, (1994).

[28] Horn, J., Nafploitis, N., Goldberg, D., “A Niched Pareto Genetic Algorithm for Multi-Objective optimization”, *In Proceedings of the First IEEE Conference on Evolutionary Computation* , 82-87, (1994).

[29] Zitzler, E., Tieler, L., *An Evolutionary Algorithm for Multi-Objective Optimization: The Strength Pareto Approach*, Technical Report 43, Zürich, Switzerland, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, (1998).

[30] Knowles, J. D., Corne, D. W., “Approximating the Non-Dominated Front Using the Pareto Archived Evolution Strategy”, *Evolutionary Computation Journal*, 8 (2), 149-172, (2000).

[31] Veldhuizen, D. V., *Multi-Objective Evolutionary Algorithms: Classifications, Analyses and New Innovations*, Ph. D. Thesis, Dayton, OH, Air Force Institute of Technology, Technical Report No. AFIT/DS/ENG/90-01, (1999).

[32] Goldberg, D. E., Korb, B., Deb, K., ”Messy Genetic Algorithms: Motivation, Analysis and First Results”, *Complex Systems*, 3(5), 493-530, (1989).

[33] Cho H., Chen Y., “Finding the ε-user Equilibrium Solution Using an Augmented Frank-Wolfe Algorithm”, *Networks and Spatial Economics*, Paper in Press, (2009)

[34] Altinel, İ., Aras, N., Oommen, B. J., “Fast, Accurate and efficient Solutions to the Hamiltonian Path Problem Using Neural Approaches”, *Computers & Operations Research*, 27, 461-494, (2000).

[35] LeBlanc, L. J., “An Algorithm for the Discrete Network Design Problem”, *Transportation Science*, 9, 183-199, (1975).

BIOGRAPHICAL SKETCH

Erdem Gündoğdu, the candidate of Master of Science in Industrial Engineering Department in Galatasaray University, was born in 1986 in İstanbul. He graduated from Kağıthane Anatolian High School in 2004. In the same year, he started his bachelor education in Industrial Engineering in İstanbul Kültür University with full scholarship. He started his second major, Economics, in 2005. In 2008, he completed his education in both majors with the second highest GPA and graduated from İstanbul Kültür University. He was rewarded as “Honor Student” in both majors. He has two proceedings of which one is presented in 23rd European Conference on Operational Research held in Bonn. The other one which is based on his final project was presented in YA/EM 2008.