PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Tesis de Doctorado
en Informática

Models and algorithms for the optimal design of bus routes in public transportation systems

Antonio Mauttone

mauttone@fing.edu.uy

Supervisor: María E. Urquhart
Orientadores de tesis: Héctor Cancela y María E. Urquhart

Tribunal: Michel Gendreau, Juan Carlos Muñoz (revisores),
Gustavo Betarte, Ángel Marín, Omar Viera

Marzo de 2011

Models and algorithms for the optimal design of bus routes in public transportation systems
Mauttone, Antonio
ISSN 0797-6410
Tesis de Doctorado en Informática
Reporte Técnico RT 11-06
PEDECIBA
Instituto de Computación – Facultad de Ingeniería
Universidad de la República.
Montevideo, Uruguay, marzo de 2011

Resumen

En esta tesis se estudian modelos y algoritmos para el diseño óptimo de recorridos de buses en sistemas de transporte público urbano colectivo. El problema conocido como TNDP (Transit Network Design Problem) consiste en determinar el número y el itinerario de líneas de transporte público y sus correspondientes frecuencias, en términos de una infraestructura dada de calles y paradas. Las soluciones deben satisfacer una demanda origen-destino dada y deben tener en cuenta los intereses de los usuarios y de los operadores y un conjunto dado de restricciones físicas, políticas y de presupuesto.

Se propone una formulación explícita de programación lineal entera mixta, que incorpora el tiempo de espera y la existencia de múltiples líneas en el comportamiento de los pasajeros. Seguidamente se discute el impacto en la estructura del modelo, al agregar restricciones de transbordos y de capacidad de la infraestructura y de los buses. El modelo se aplica (usando un solver estándar) a casos de prueba muy pequeños, así como a uno real relativo a una ciudad pequeña que consta de 13 líneas de buses.

Con el propósito de atacar casos de mayor tamaño, se propone un algoritmo constructivo ávido que produce un conjunto de recorridos que son convenientes tanto para los usuarios como para los operadores, teniendo en cuenta restricciones de transbordos. Utilizando un caso de prueba real, se muestra que el algoritmo propuesto mejora resultados del estado del arte.

Como una extensión del algoritmo constructivo, se representa la existencia de los objetivos en conflicto de usuarios y operadores usando un modelo de optimización combinatoria multi-objetivo para el TNDP. Este nuevo modelo se resuelve con una metaheurística que explota la naturaleza multi-objetivo del problema para resolverlo eficientemente. Utilizando un caso de prueba de referencia existente en la literatura y uno real, se muestra que el algoritmo propuesto mejora resultados del estado del arte y produce soluciones de características comparables a las del sistema real.

Los valores objetivo del algoritmo constructivo y de la metaheurística se comparan con valores correspondientes a soluciones de referencia; en el primer caso se compara contra soluciones óptimas obtenidas con la formulación matemática, mientras que para el segundo se compara contra la solución que opera el sistema de transporte público de la ciudad correspondiente al caso de prueba real.

Finalmente se discuten las relaciones entre las diferentes contribuciones de esta tesis y se comentan varias cuestiones relacionadas a la aplicación de las metodologías propuestas a casos reales. También se formulan algunas opiniones y recomendaciones en relación a futuros desarrollos de éste tópico de investigación.

Palabras clave: Transporte público, Diseño óptimo de recorridos y frecuencias de buses, TNDP, Programación lineal entera mixta, Heurísticas, Caso de prueba real.

Abstract

In this thesis we study models and algorithms for the optimal design of bus routes in urban public transportation systems. The problem known as TNDP (Transit Network Design Problem) consists in determining the number and itinerary of public transportation lines and their corresponding frequencies, in terms of a given infrastructure of streets and stops. The solutions should satisfy a given origin-destination demand and should take into account the interests of users and operators and a given set of physical, policy and budgetary constraints.

We propose an explicit mixed integer linear programming formulation which incorporates the waiting time and the existence of multiple lines in the behavior of the passengers. Then, we discuss the impact in the structure of the model of adding transfer, infrastructure and bus capacity constraints. We apply the model (using a standard solver) to very small test cases as well as to a real one, related to a small-sized city comprising 13 bus lines.

In order to deal with cases of larger sizes, we propose a greedy constructive algorithm that produces a set of routes that are convenient for both users and operators, taking into account constraints related to transfers. By using a real test case, we show that the proposed algorithm improves results from the state of the art.

As a further extension, we represent the existence of the conflicting objectives of users and operators using a multi-objective combinatorial optimization model for the TNDP. This new model is solved by a metaheuristic that exploits the multi-objective nature of the problem in order to solve it efficiently. By using a benchmark test case and a real one, we show that the proposed algorithm improves results from the state of the art and produces solutions with characteristics comparable to the real one.

Objective values of both constructive and metaheuristic algorithms are compared with values corresponding to reference solutions; for the first one we compare against optimal solutions obtained with the mathematical formulation, while for the second one we compare with the solution operating the public transportation system of the city corresponding to the real test case.

Finally we discuss the relationships between the different contributions of this thesis and we comment several issues related to the application of the proposed methodologies to real cases. We also give some opinions and recommendations concerning future developments in this research field.

Keywords: Public transportation, Optimal design of bus routes and frequencies, TNDP, Mixed integer linear programming, Heuristics, Real test case.

Contents

List of Figures

List of Tables

List of Symbols

Symbol	Defined on page
α	60
β	20
Δ_s	36
δ_k	29
ϵ	42
η	73
Θ	34
θ	34
κ_e	37
Λ_{er}	37
λ	76
λ_1	77
λ_2	77
ρ_{max}	55
σ_t	73
τ	36
Φ_k	72
ϕ_k^*	73
ω	23
A	16
A^D	31
A_i^D	36
A^T	18
A^V	31
A^W	31
A_n^-	31
A_n^+	31
A_n^{W+}	32
B	33
b_n	31
c_a	16
c_e	17
D	16
$D_0(R)$	52

Symbol	Defined on page
$D_0(S)$	73
D_0^{min}	52
$D_{01}(R)$	52
$D_{01}(S)$	73
D_{01}^{min}	52
D_k	29
D_{tot}	52
d_{ij}	16
E	17
E_r	33
f_a	31
$f_{(a)}$	34
f_{max}	73
f_{min}	73
f_r	33
G	16
G^T	18
K	29
L	42
l	55
N	16
N^C	16
N^P	16
N^S	16
N^T	18
N_S	77
O_k	29
$\mathcal{P}$	73
R	33
$r_{(a)}$	33
rcl	76
$t_{ij}(R)$	51
t_k	52
t_{max}	55
t_{max}^{end}	78
t_{max}^{ini}	78

Symbol	Defined on page
tt	72
tv	72
tv^*	82
tw	72
tw^*	82
t^*_{ij}	51
$\bar{U}$	82
U^*	83
V	17
V_n	31
v_a	20
w_n	32
x_a	31
x_r	33
Y_1	51
Y_2	51
y_{rf}	34
Z_1	72
Z_2	73

Acknowledgements

I would like to thank several people and institutions that have made possible the development of this thesis.

First of all I would like to express my gratitude to my supervisors Héctor Cancela and María E. Urquhart for their guidance and support during the whole development of this thesis. I would like to thank to Martine Labbé and Rosa Figueiredo, for their guidance during my stay at the Département d'Informatique of Université Libre de Bruxelles and to Ricardo Giesen for his guidance during my stay at Departamento de Ingeniería de Transporte y Logística of Pontificia Universidad Católica de Chile. Also I would like to thank the anonymous referees for their comments and suggestions concerning submitted papers and projects and to people who participated during my talks at different conferences.

Several institutions have made possible the development of this thesis through financial support in specific projects, programs or grants. I would like to thank to Comisión Académica de Posgrado of Facultad de Ingeniería, to Comisión Sectorial de Investigación Científica of Universidad de la República, Programa de Desarrollo Tecnológico of Dirección Nacional de Ciencia y Tecnología and Programa de Desarrollo de las Ciencias Básicas. Also to Programme ALFA II-0457-FA-FCD-FI-FC and LACCIR Short Stays Program that made possible my academic stays on Belgium and Chile respectively.

I thank the Municipality of Rivera for its collaboration concerning the data needed for constructing the real test case and the Instituto de Computación for its flexibility that enabled me to concentrate on tasks related to the thesis at different periods of these years.

Finally, I would like to thank to all those have worked in the different projects related with this thesis, in particular the undergraduate students of Computer Engineering. Also I thank to my mates of Instituto de Computación, friends, family and specially to Ana.

Chapter 1

Introduction

This thesis is about models and algorithms for the optimal design of routes (or lines) in urban public transportation systems (hereafter, public transportation systems). In most cities of the world there are public transportation systems, either conceived as a service that should be provided to the inhabitants (like electricity or drinking water), as a tool for urban planning (used to guide land use or to alleviate street congestion), or as business of private companies.

A public transportation system is composed by an infrastructure and services that operate over it. Those services are provided to persons that need to travel along the city (hereafter, users). When designing and implementing the system, two types of monetary costs arise: fixed costs due to construction of the infrastructure and variable costs due to operation of the services. These costs are perceived by the whole society; in the most common case, the government builds the infrastructure, the operators (companies) provide the services and the users pay a given fare to access such services. There is another cost which is not monetary and is perceived by the users: the travel time. The total cost of the system is the sum of the monetary fixed and variable costs plus the travel time of all users.

When designing a whole public transportation system (either from an existing one or from scratch), many decisions should be taken, which impact on the total cost of the system. That process of decision making faces the existence of many feasible alternatives which result in different levels of total cost. The design of a public transportation system can be modeled as an optimization problem, in particular as a cost minimization one. However, such a problem is intractable as a single monolithic unit, given the number of variables, relations among them and even conflicting objectives. For this reason the problem is divided in parts of smaller size, in such a way that the resulting problems can be tractable. Usually that division corresponds to the different scopes (and their internal organization) where decisions are taken [98]. Also, the subproblems belong to the different planning stages of the whole system, defined according to the time horizon where they take place, namely strategic (long term), tactical (medium term) and operational (short term) [33].

Among the different existing technologies to construct and operate a public transportation system, this work is restricted to systems based on buses, where we assume an already available infrastructure (street network, stops). Therefore we do not consider de-

cisions related to building new infrastructure (for example, exclusive bus corridors, tram or underground networks); this implies that fixed costs are not part of our models. The subproblems that are tackled during the planning of a public transportation system based on buses, according to the division proposed in [18] are:

1. Design of routes, where one should decide the number of lines and the itinerary of each one in terms of the street network, in such a way that the demand of travel between different points of the city is satisfied.

2. Frequency setting, where the time interval between buses of each line is decided. Usually these decisions are taken for different scenarios of demand, for example, different seasons of the year or times of day.

3. Timetable construction, where one should determine the exact starting and ending time of each bus performing every route of every line.

4. Fleet assignment. Given a timetable, it should be determined the sequence of trips assigned to each bus, respecting constraints of available fleet and depot location.

5. Crew assignment. Drivers and other staff needed to operate each bus should be assigned, respecting the working rules.

In most public transportation systems, problem 1 is solved within the scope of the municipality or planning agency (hereafter, regulator), while problems 2 and 3 are solved jointly by the regulator and the operators. Problem 1 corresponds to strategic planning while problems 2 and 3 correspond to tactical planning. In these problems, the main objective is to design a system which offers the highest possible level of service, with the lowest possible cost. From the point of view of the users, such a system should satisfy the needs of travel of all the inhabitants of the city, with the lowest possible travel time and fare and reasonable comfort conditions. However, there are different constraints that preclude the existence of such system: capacity of the infrastructure and buses, available budget and other constraints resulting from the fact that the public transportation system is part of a more complex one: the city where it is embedded. For these reasons, the regulator should take into account all these elements when designing routes, frequencies and timetables. On the other hand, problems 4 and 5 (corresponding to operational planning) are solved in the context of the operators, where usually there is a single objective: cost minimization. The five problems mentioned above are solved sequentially, therefore decisions taken in a given stage of the sequence are conditioned by decisions taken in previous stages. Although the division proposed in [18] suggests an ordering for solving the subproblems, that division is not the only possible one [91]. At first sight, one may think that independently of the division in stages and their sequencing, it is desirable to solve optimally each one of the associated subproblems, thus contributing to the optimization of the overall public transportation system. Actually this is not true, since there are examples which show that by solving problems 4 and 5 simultaneously we can obtain better results than if we solve them separately [50]. However, the current state of practice shows that it is hard to solve simultaneously problems that involve decisions taken in different scopes; clearly this is not the case of problems 4 and 5, which are both solved within the same scope. Therefore, to solve problems 1, 2 and 3, the usual way is to consider a primary objective

to be optimized, while taking care in other desirable properties of the solutions which contribute indirectly to the optimization of 4, 5 and the overall system.

1.1 Optimization of routes

The problem of route optimization for a public transportation system is formulated in terms of a graph, whose nodes represent intersections of streets or zones and whose arcs represent connections between such nodes (for example, a street segment between two consecutive intersections or a connection between two adjacent zones). The correspondence between elements of the graph model and the reality depends on the level of aggregation adopted. A given origin-destination matrix expresses the number of trips that should be satisfied (each one to be performed by one person) between nodes of the graph in a specific period of the day. A route is a sequence of adjacent nodes in the graph. The term *line* is used in this work as synonymous of route, however in real systems, a line (usually identified by a number or name from the point of view of the users) can be composed by various routes (for example, forward and backward directions).

In general terms, the problem of route optimization is a variant of the generalized combinatorial optimization problem of network design [76]. In that problem one should select a subset of arcs from a given set, considering aspects like level of service, fixed and variable costs, structural, physical and behavioral considerations (among others); these aspects may be modeled either in the objection function or as constraints. The main difference of the problem of route optimization for public transportation systems with respect to the general network design problem is that in addition to determining which arcs to include in the solution, one should determine how these arcs are combined to form different routes. Also one should determine the frequency of each route [33], since that variable has direct impact in the level of service to the users and in the cost of the operators. Given that we are considering as main objective to maximize the level of service, one should optimize a performance measure (or several ones) of a set of routes, from the point of view of the users. Typically these measures are the travel time (a component of the cost) and the occupancy of the buses (as a measure of comfort). The first one usually has three components: access and egress time that represent the walking time to reach the origin stop and the final destination of the trip respectively, waiting time at the stop and on-board (inside the bus) travel time; also a transfer time between lines may need to be considered. The occupancy of buses is expressed in terms of the demand (amount of persons) that use each line in relation to the provision of such resource. The computation of these measures require to model the perception that the users have of the "goodness" of a set of lines; for doing that, one should "simulate" the use of the lines by the users. This problem is modeled by a sub-model called the *assignment model*, which distributes the demand over a given set of routes, assigning flows that represent how the users use the lines. The assignment model should apply the hypothesis about the behavior of the users with respect to a set of lines. The assignment problem has been studied as part of the route optimization problem [70, 77, 109], the frequency optimization problem [17, 60] or as an isolated problem [22, 30, 110]. In this thesis we do not contribute to this topic, however the assignment model is considered as a central component of any model of route optimization, given that usually it determines the computation of the objective function.

The following hypothesis delimitate the scope of this thesis, concerning the models and algorithms studied for route optimization:

1. The public transportation system is considered in isolation from other modes of transportation, for example private cars. Moreover we consider a single mode of public transportation, which is based on buses.

2. We do not consider the interactions between the public transportation system and the dynamics of land use of the city where it is embedded.

3. The demand is considered inelastic. We assume a fixed set of users that do not have other alternative for traveling (captive clients).

4. We do not model the impact that might cause the fare charged for using the service of public transportation, in the behavior of the users concerning the use of the lines. It is known that different fare structures have consequences in such behavior [124].

5. We do not consider the existence of advanced traveler information systems (ATIS), which also have influence in the behavior of the users [97].

6. We assume that users are sensitive to the waiting time and to transfers. It is known that certain features of some systems, like special infrastructure (bus stops or stations) and operation schemes (high frequency, coordinated timetables, ATIS), contribute to decrease the negative perception that users have about waiting time and transfers.

1.2 Literature review

In this section we review the existing literature about optimization of routes for public transportation systems. We focus on the aspects related to mathematical models and algorithms to solve the problem. Similar reviews with the same focus have been published in [33, 98]. Other aspects of the problem have been considered in [59, 68].

Most works related to models and algorithms for route optimization in public transportation systems are approximate methods (heuristics), based on formulations that are not explicit. A formulation is said to be explicit if it has completely defined all the mathematical expressions that represent decision variables, constraints and objective function. In some cases the problem is decomposed in subproblems that have an explicit formulation; the decomposition usually consists in solving firstly the route design and then the frequency setting.

In [109], a zonal division of the city and a graph that contains information about on-board travel time is considered. Based on this information, a set of routes is determined by evaluating different "skeletons" (using a procedure inspired from [70]); a skeleton is a sequence of zones whose extremities are terminals (previously identified). In a second stage, a minimization problem is considered, whose objective function includes the total travel time (waiting, on-board and transfer) and a factor that penalizes the number of standing passengers. In addition, a constraint on the fleet size (buses circulating simultaneously in the system) is considered; the fleet size is obtained from the duration of each

route and its frequency. The assignment model assumes that passengers traveling between two nodes of the graph are distributed among the fastest lines that connect both nodes. The resulting minimization problem is solved by applying a gradient projection method.

In [35], a first procedure determines the streets of the city where the lines of public transportation will be defined, by solving approximately a network design problem [76]. In a second stage, a procedure creates routes until covering the whole demand, creates new routes based on identified transfer points and finally combines routes and eliminates parts of routes with low utilization. In the following stage, a frequency optimization problem similar to the one considered in [70, 109] is solved.

While the studies mentioned above generate a set of routes with frequencies from an empty set, in [77] a procedure to improve an existing system is proposed; it is based on operations of insertion and elimination of nodes in routes and interchange of parts between different routes. Afterwards, a model that assigns a fixed fleet of buses to the resulting routes is applied, so as to minimize the waiting time. The assignment model used is inspired from [22].

In [18], another model based on two stages is proposed. In the first one, an objective function is minimized, which contains the deviation of the on-board travel time of a given solution (set of routes) with respect to the time of the minimum cost path in the graph that represents the streets of the city (independently of any set of routes); in addition, the total transfer time is computed. The constraints include limits on route length and number of routes. In the second stage, two terms are added to the objective function considered in the previous stage: the total waiting time and the fleet size. The last term represents the interest of the operator, therefore the model has multiple objectives, in this case included into the same objective function using weighting factors. This work proposes an algorithm of route generation based on a breadth first search in the street graph, that can be used as an auxiliary subroutine to solve the proposed model.

A model based in [18] is proposed in [65], putting special emphasis in the multi-objective characteristic of the problem. This approach seems to be reasonable, given that in the context of strategic planning, limit values in the single objectives (for example fleet size) may not be known beforehand. Instead, the regulator may be interested in exploring different alternative solutions with different levels of compromise (or trade-off) between the objectives of users and operators. To obtain such solutions efficiently, an algorithm specially designed for this purpose is developed.

In [10], a greedy algorithm for generating routes is proposed; it produces solutions that satisfy constraints of demand covering, introduced by this work. These constraints state that a given proportion of the total demand should be covered with no more than a given number of transfers; this introduces realism to the model. Demand covering constraints have been treated indirectly in previous works by limiting the transfers implicitly, for example by including them in the objective function. In a related work [9], the authors combine this route construction algorithm with an improvement procedure and an assignment model [8] based on [60], to solve the route optimization problem considering constraints of bus capacity.

All the studies mentioned above have done the main contributions to the modeling of the problem. The solution methods used are heuristics with different degrees of knowledge of the real problem; in some cases, solution improvement procedures are proposed, but

concepts related to local search and neighborhoods are not explicitly mentioned. More recently, several studies which explore new solution methods for the existing models and variants have been published. The salient characteristic of these methods is the use of metaheuristics [14, 56] as an optimization tool. Most of these recent studies about route optimization for public transportation systems use already existing assignment models. Therefore the latest contributions in the assignment aspect are located in the bibliography specific to this topic; updated surveys can be found in [33, 69].

In [101], the authors propose a genetic algorithm to solve a variant of the model of route optimization proposed in [9], using the assignment model presented in [8]. The algorithm selects the "optimal" subset of routes from a pool of many possible routes, generated by using an algorithm similar to the one proposed in [18]. In this work the frequencies are not determined by the genetic algorithm, while in [113] they are codified as part of the solution, therefore they are also "optimized".

In [94], a genetic algorithm which uses several genetic operators that include problem knowledge is proposed. The used assignment model assumes that the users select a priori a single line for traveling from the origin to the destination; this hypothesis is not realistic, since a user having multiple lines with identical on-board travel time, usually will take the first one passing by the stop. The objective function of the optimization model has an explicit formulation once the values of the assignment variables are known for each feasible solution; this allows to compute optimal frequencies for that solution.

A stream of work considers the application of different metaheuristics to solve a variant of the optimization model proposed in [9]; the authors apply Tabu Search [44], Simulated Annealing [43] and Genetic Algorithms [42]. These algorithms are based on a very big pool of routes, generated using an algorithm that computes the k minimum cost paths [122] between pairs of nodes of the street graph. Where a neighborhood structure is needed, the neighbors are obtained by changing one route r in the current solution, by a route that is contiguous to r in the pool of routes. Two routes are contiguous in the pool if they have subsequent indexes in the list of k minimum cost paths, considering the list sorted with respect to the cost of its elements.

Usually the algorithms that tackle the route optimization problem using metaheuristics require repeated invocations to the subroutine that implements the assignment model. This component is critical and the consumption of computational resources is significative with respect to the whole running time of the optimization algorithm. This is because the assignment algorithms imply searching and enumerating paths in the solution composed by routes. To overcome this difficulty, in [1] a parallel version of the algorithm proposed in [113] is implemented, dedicating a computational unit to the execution of the assignment model.

More recently, explicit mathematical programming formulations for the route optimization problem have been proposed; in all cases the formulations are of mixed integer linear programming (MILP) type.

In [120], a model that minimizes operator's cost under constraints of bus capacity and number of routes is proposed. The formulation has the ability of constructing routes from edges of the graph that represent the streets; for doing that, a big number of auxiliary variables is introduced, however the total number of variables and constraints is polynomial. The formulation is solved applying directly a MILP solver.

The formulation proposed in [58] minimizes an objective function that combines line cost, number of transfers and on-board travel time. The first term represents the interest of the operators, while the other terms represent both the interest and the behavior of the users. The different terms are weighted by coefficients that should be determined. The constraint set includes street capacity, line length and maximum number of transfers. The formulation is solved applying directly a MILP solver, over a set of predetermined lines.

In [107], the authors propose a multi-commodity formulation which minimizes the on-board travel time and the number of transfers, under constraints of bus capacity and budget. To model transfers, a particular graph structure is used, which increases the size of the model significantly. The number of variables of the formulation is super-polynomial, therefore the decomposition of Dantzig-Wolfe [28] is used to solve the linear relaxation.

In a similar vein, the model proposed in [15] minimizes the on-board travel time and the operator's cost, under constraints of bus and street capacity. In this model, two conflicting objectives are included into the same objective function and the transfers are ignored. The number of variables of the formulation is super-polynomial, therefore a column generation method is used to solve the linear relaxation; then a feasible integer solution is obtained using a heuristic procedure.

These last two studies [15, 107] prove that their respective formulations state problems that belong to the NP-hard class. Moreover, the used assignment models are implicit in the formulations and they model situations that are not realistic in many scenarios related to public transportation systems based on buses; in particular they do not model the waiting time in the behavior of the users. It is worth mentioning that in these cases where an explicit mathematical formulation is used to model the problem, either (global) optimal solutions or solutions with a lower bound (as a reference of distance to optimality) are reported. None of the methods mentioned on this review which are based on heuristics without an explicit formulation, provide this type of result; as an exception, a lower bound for the on-board travel time component of the objective function is reported in [9].

The test cases used in all the studies mentioned above can be classified on three types: (1) abstract cases usually small-sized, (2) cases corresponding to real cities small or medium-sized and (3) cases corresponding to real cities of big size. Example of type 1 is the case used in [65] whose network has 8 vertices and 13 edges, while example of type 2 is the case relative to the city of Postdam, Germany used in [15] whose network has 410 vertices and 891 edges. As example of type 3 we mention the case used in [1] relative to the city of New Delhi, India, with 1332 vertices and 4076 edges. It is worth mentioning that the size of the graph is not directly related to the size of the city that it represents, since it strongly depends on the level of aggregation of the data and the procedure used to construct the case for the model of route optimization. It should be taken into account that not only the size of the graph determines the computational cost of an algorithm to solve this problem; also the number of nonzero elements of the origin-destination matrix plays an important role in determining that cost. Among all the published test cases for the problem, only the one proposed by Mandl [78] has been used as benchmark by several authors [9, 20, 40]; part of the information of this case has been added recently to the repository OR Library [75].

Finally we should mention on this review, several studies which are related to the topic of this thesis, which apply to variants of the problem or to problems that are closely

related to the route optimization for public transportation systems. In [46, 47, 121], the interaction of the public transportation mode with other modes is modeled, while in [61, 72], the elasticity of the demand is modeled within the model of route optimization. In [21, 40, 80, 119], the problem of route optimization is solved without considering the frequencies as decision variable; in all these cases the assignment model can be simplified. The problems of stop location [63], design of limited-stop services [73] and frequency optimization [26] share an important characteristic with the problem of route optimization: all of them require an assignment sub-model that represents the behavior of the users with respect to a set of lines of public transportation.

1.3 Motivation of this thesis

The main motivation of this thesis is the study of the problem of route optimization in public transportation systems, from an Operations Research perspective. The goals are to develop models and algorithms that can be applied to real cases related to public transportation systems based on buses. The literature review presented in Section 1.2 allows to identify the following specific topics where interesting research can be done:

- The combinatorial characteristic and the need of representing the behavior of the users are the main difficulties to formulate and solve the problem. The existence of multiple objectives should also be taken into account. These issues have been already identified by other authors in previous studies [9, 20]. However, the influence of capacities over the behavior of the users has not been widely discussed in the context of this problem. A relevant issue that must be taken into account is that when bus capacities are introduced, optimizing a single measure that represents the various actors of the problem (users, operators, regulator), usually will lead to results that are not consistent with the behavior of the users [93].

- The existing mathematical programming formulations do not model (into the same formulation) realistic scenarios under some of the hypothesis stated in Section 1.1, because they do not include the waiting time in the behavior of the users and they do not control the number of required transfers in the optimal solution.

- The methods based on heuristics without an explicit formulation do not provide an evaluation of distance to optimality of the produced results.

- The computational experiments with cases related to real cities do not always present details of the construction of the case (graph and origin-destination matrix), therefore the interpretation of the results in terms of the reality is difficult.

Given that we are dealing with a real problem which has direct impact to the society, it is worth asking about the feasibility of application of the results of this research. Although the literature published concerning the optimization of routes for public transportation systems has grown in the last years, the tools used in practice to solve this problem are much more scarce than their counterparts related to assignment models for public transportation systems. Possible reasons for this fact are:

- According to [98], there is a tendency to keep stable the design of routes of public transportation in any city, mainly due to the impact that changes may have on the users of the system.

- An optimization model is usually conceived as a normative tool, in the sense that it produces a recommendation of decision to be taken. A set of routes completely generated by an automatic tool based on a mathematical model is likely to be taken with skepticism by the planner (who takes decisions in the scope of the regulator, also referred as decision maker). The lack of trust in any model of route optimization in some cases may be due to the perception that it is impossible to include all the aspects of the problem and the knowledge of the planner into the model. On the other hand, the assignment models are descriptive tools, in the sense that they provide a picture of the use of the system by the users; these type of tools are used as decision support systems in situations where the planner wants to evaluate quantitatively the impact of different given alternative designs.

- The data necessary to apply a model of route optimization, in particular those related to the demand, are very costly to obtain and they are subject to errors. Given that the demand information is a critical input for the models, there is a risk that errors in data propagate through the model, causing that the solution obtained is not useful because it corresponds to a distorted scenario. Although the presence of errors on data might justify the lack of trust in the results produced by the models, the existing applications of descriptive tools (that use demand data intensively) in the planning of public transportation systems are large; several commercial software packages that include assignment models for public transportation (for example: EMME [64], ESTRAUS [48] and VIPS [102]) have been used as decision support systems in many cities all over the world.

- The models of route optimization for public transportation systems are difficult to formulate and solve.

In [98] (year 1994), the authors suppose that automatic methods for generating routes will not be used in the short term. However, in subsequent years the studies published concerning models and algorithms for route optimization in public transportation suggest that there is still interest (and need) of expanding the state of the art in this topic. There is an underlying idea that it is always desirable to "facilitate" the work of the planner, with more or less degree of automatization. In recent publications, new advances in the mathematical modeling of the problem can be found, as well as new algorithmic techniques (exact combinatorial optimization methods, metaheuristics) and applications to real cases in different places like Chile [47], Germany [15], Hong Kong [112], Italy [23], Spain [100] and United States [123].

1.4 Contributions of this thesis

The main contributions of this thesis are the following:

- An explicit mathematical formulation for the problem.
- Approximate algorithms (heuristics) that solve some aspects of the problem.
- Application of the proposed models and algorithms to a real case and discussion of the results.
- Implementation of a software tool that allows to apply the methodologies developed to real cases related to small and medium-sized cities.

The developed methodologies approach the problem without requiring special local knowledge of the reality where they will be applied, i.e., only the data required by the models are needed. Thus, the methodologies are somehow generic and they are not strongly biased towards a particular reality. The results are compared quantitatively, using indicators that measure the performance of the solutions obtained (for example, travel time or fleet size); we do not incorporate the judgement of the planners, neither in the generation of solutions nor in their evaluation. This thesis does not propose advances concerning the assignment model, however we have used such models that are realistic under the hypothesis stated in Section 1.1. The models and algorithms studied are conceived to be used either in the context of strategic or tactical planning of a public transportation system; they can be used to define a completely new set of routes for the system, or to make changes or adjustments to an existing one.

Concerning the mathematical modeling, we have developed an explicit mathematical programming formulation that includes the following aspects of the problem: (i) the behavior of the users, taking into account the waiting time and multiple routes, (ii) constraints of minimum percentage of demand satisfied with a given number of transfers (demand covering constraints) and (iii) the capacity of the infrastructure and the buses. The consideration of these three aspects into the same formulation is a contribution to the state of the art, since the existing formulations either do not model all these aspects or they are not explicit. The proposed formulation allows to reason about the mathematical structure of the problem; in particular it is useful to illustrate the difficulty of solving the problem when including the demand covering and the infrastructure and bus capacity constraints. For very small instances and particular cases of the problem, the proposed formulation allows to obtain the optimal solution, using standard MILP techniques. Thus, the formulation is useful to evaluate the distance to the global optimum of the solutions produced by the approximate algorithms. The formulation is also applied to a real case related to a small-sized city.

Concerning the development of algorithms, the contributions are in two directions: (i) a greedy algorithm to construct a set of routes, which takes into account both objectives of users and operators and demand covering constraints, and (ii) an algorithm based on the GRASP metaheuristic [45] that solves the model of optimization of routes and frequencies proposed in [9] with a multi-objective approach. The obtained numerical results improve the state of the art. Both algorithms are applied to a real case, showing that they have

practical usefulness. The first algorithm has a structure that allows to be used in an interactive way by the planner while the second one is designed to facilitate the planner the task of dealing with the multi-objective aspect of the problem.

The developed algorithms have been applied to a real case of study relative to Rivera, a small city of 65,000 inhabitants in Uruguay. The procedures to obtain the data and to construct the case were performed in a project which is strongly related with this thesis. This is an important point, since allow us to keep control over all the aspects of the research concerning this thesis, in this case, the data collection and processing. The results produced by the algorithms are discussed in light of the hypothesis of the optimization model and the heuristic nature of the solution method. Moreover, they are compared against the solution corresponding to the public transportation system of Rivera, taking into account the source of the data and the procedures used to construct the case.

Finally we have developed a software tool that allows to prepare the information needed to construct a test case for the application of the route optimization algorithms. The tool enables the researcher to experiment with new algorithms of optimization and evaluation of routes and facilitates the data processing and the analysis of solutions (set of routes). The software also allows to communicate the results to the planner, by using a graphical user interface and a data format that is standard in the geographical information systems that are usually available. This is an important requirement looking at the application of the developed methodologies to real cases.

The research work undertaken during the development of this thesis has been presented and published in the following instances:

Reviewed journals

- A. Mauttone and M. Urquhart. A multi-objective metaheuristic approach for the Transit Network Design Problem. Public Transport 1(4):253-273, 2009.
- A. Mauttone and M. Urquhart. A route set construction algorithm for the Transit Network Design Problem. Computers & Operations Research 36(8):2440-2449, 2009.

Technical reports

- A. Mauttone. Formulación de programación matemática para el problema de optimización de recorridos y frecuencias en sistemas de transporte público. Reportes Técnicos PEDECIBA Informática, Instituto de Computación de la Facultad de Ingeniería, RT 09-14, 2009.

Conferences

Complete papers:

- R. Alvarez, M. Martínez and A. Mauttone. Heurística de búsqueda de entorno variable para el problema de ruteo de transporte público urbano. 42º Simpósio Brasileiro de Pesquisa Operacional, Bento Gonçalves, Brazil, 2010.
- A. Mauttone and M. Urquhart. Una metodología para la optimización de recorridos y frecuencias en transporte público y su aplicación a un caso de estudio real. 7º Congreso de la Vialidad Uruguaya, Montevideo, Uruguay, 2009.

- A. Mauttone and M. Urquhart. Optimización multi-objetivo de recorridos y frecuencias en transporte público aplicado a un caso de estudio real. XIII Congreso Chileno de Ingeniería de Transporte, Santiago, Chile, 2007.
- A. Américo, F. Martínez, A. Mauttone and M. Urquhart. Multi-objective evolutionary algorithm for the transit network design problem. VI International Conference on Operational Research for Development, Fortaleza, Brazil, 2007.
- A. Mauttone and M. Urquhart. A Multi-Objective Metaheuristic approach for the Transit Network Design Problem. 10th International Conference on Computer-Aided Scheduling of Public Transport, Leeds, United Kingdom, 2006.

Extended abstracts:

- A. Mauttone, M. Labbé and R. M. V. Figueiredo. A Tabu Search approach to solve a network design problem with user-optimal flows. VI ALIO/EURO Workshop on Applied Combinatorial Optimization, Buenos Aires, Argentina, 2008.
- A. Mauttone and M. Urquhart. Modelo de programación lineal entera para el diseño de redes de transporte público. XIV Congreso Latino Ibero Americano de Investigación de Operaciones, Cartagena de Indias, Colombia, 2008.
- A. Mauttone and M. Urquhart. Una heurística basada en memoria para el problema del diseño de recorridos en transporte público urbano. XIII Congreso Latino Iberoamericano de Investigación Operativa, Montevideo, Uruguay, 2006.

Abstracts:

- A. Mauttone, R. Giesen and M. Urquhart. Formulation and Heuristic Solution for the Transit Network Design Problem. XV Congreso Latino Ibero Americano de Investigación de Operaciones, Buenos Aires, Argentina, 2010.
- A. Mauttone and M. Urquhart. GRASP for Multi-Objective Optimization of Public Transportation Networks. XV Congreso Latino Ibero Americano de Investigación de Operaciones, Buenos Aires, Argentina, 2010.
- A. Mauttone, R. Giesen and M. Urquhart. Transit Network Design Problem: A Mathematical Formulation and Heuristic Solution. Transportation and Logistics Workshop, Reñaca, Chile, 2009.
- H. Cancela, A. Mauttone, M. Urquhart and O. Viera. Models and algorithms for the Transit Network Design Problem. Transportation and Logistics Workshop, Reñaca, Chile, 2009.

1.5 Structure of the document

This document is organized as follows. Chapter 2 presents the theoretical background including concepts, definitions and notation used in the thesis; it states formally the problem to be studied. A specific section is dedicated to the assignment model. In Chapter 3

we present the proposed mathematical programming formulation for the problem of route optimization in public transportation systems; starting from a base formulation we then include constraints of demand covering and infrastructure and bus capacity, studying the implications that they pose. A section of numerical results is presented, that applies the proposed formulation to different test cases including a real one. Chapter 4 presents the greedy algorithm proposed to construct a set of routes; the reported experiments make comparisons with results from state of the art and show that the algorithm has desirable properties to be used as subroutine of an algorithm that optimizes routes and frequencies. In Chapter 5 the multi-objective approach to the problem is presented, as well as an algorithm to solve approximately the proposed model; results are compared with results from state of the art and experiments show the usefulness of the approach in a real case. A dedicated section of the experiments discusses the results obtained with the real case in comparison with the real solution (the one operating the public transportation system of the city). Finally, in Chapter 6 we elaborate conclusions from the overall research work undertaken in this thesis and we identify future work. The document includes two appendixes: in Appendix A we describe the construction of the real case related to the city of Rivera while Appendix B shows the main features of the software tool developed to assist the research concerning the optimization and evaluation of routes for public transportation systems.

Chapter 2

Background

In this chapter we present the concepts, definitions and part of the notation used in the remaining part of the document. Some additional notation is defined in specific chapters. We present the modeling of the problem of route optimization in terms of graphs. A section is devoted to the assignment model. Finally, the problem to be studied in the thesis is defined.

2.1 Modeling the problem in terms of graphs

The modeling of the problem of route optimization for public transportation systems, requires to represent the infrastructure over which the routes will be defined and the demand that should be satisfied. Once the routes are defined, it is necessary to represent the trajectories that the users will follow from their origins to their destinations using those routes. A mathematical structure suited to represent these elements is the graph. In some cases, the elements of the graph have a direct correspondence with elements of the reality, for example, each node represents an intersection of streets and each arc represents a section of street between two intersections. In other cases, the elements of the graph represent fictitious entities, for example, zone centroids or waiting arcs. A zone centroid is a point that represents an entire geographical zone [99] and a waiting arc is used to calculate the waiting time for the users [110]. The literature presents a representation based on graphs for urban transportation problems, which is widely used [99, 108]. Moreover, the specific literature of assignment models for public transportation presents several representations using graphs, each one proposed for the particular case that is modeled [97]. The models of route optimization require a combination of different graph models, given that we need to represent into the same model, decisions about the structure of the routes (in terms of the infrastructure, in this case, street network and bus stops) and the behavior of the users (in terms of routes and walking paths). This issue is not commonly recognized in the literature.

In the following we describe a graph model proposed to represent the elements of a public transportation system, needed to formulate the problem of route optimization; the description is divided in aspects related to infrastructure and demand (Section 2.1.1), routes (Section 2.1.2) and assignment (Section 2.2).

2.1.1 Infrastructure and demand

The routes are defined in terms of the infrastructure, that in the case of this thesis is given by the streets considered for running the buses and the stops of public transportation. On the other hand, the demand is determined by the users of the system, who inhabit the city and need to perform trips from certain origins to certain destinations. The trips can originate at any geographical point of the city, in particular at households, buildings of service, education, health, etc. Given that it is not convenient to represent any possible point of trip generation, the models of urban transportation usually consider the trip generation at the level of zones [99]. Thus, the demand is considered as concentred in a fictitious point located inside the zone called centroid, which is connected to the bus stops that lie inside the zone.

We model the infrastructure and the demand using a directed graph $G = (N, A)$, where each node $n \in N$ can be of type street, stop or centroid (eventually at the same time). Let $N^S, N^P, N^C \subseteq N$ be the node sets (not necessarily disjoints) of type street, stop and centroid respectively. The arcs of the graph represent connections between the nodes, which can be of type travel (a street segment whose endpoints are of type street or stop) or walk (one endpoint is a stop and the other one is a stop or a centroid). Each arc $a = (i, j) \in A$ has an associated cost c_a that represents a travel time. For a travel arc, this cost represents the time spent by any bus for traveling from node i to node j; therefore, it is also the time that will experience the users (on-board the bus) of a line which passes by that arc. The cost of a walk arc represents the time spent by any user for walking between stops or between a stop and a centroid (the place where the trip starts or ends). Figure 2.1 presents an example of the graph model described above. A similar model has been proposed in [6].

The demand is given by an origin-destination (OD) matrix defined as $D = \{d_{ij}\}$ where i and j are centroid nodes (they belong to N^C). Each element $d_{ij} > 0$ (called OD pair) expresses the number of trips that should be satisfied by time unit in a given time horizon, from i to j; each trip will be performed by one person that will occupy one place inside the bus.

2.1.2 Routes

The physical structure of the routes is a sequence of streets where the buses will pass; in real public transportation systems there can exist forward and backward routes, circular routes and other particular structures of routes. The degree of detail of the representation of the route structure strongly depends on the degree of detail of the model given by the graph G. An important aspect of the modeling of the route structure is the inclusion of the stops. Those elements represent the points through which the users access to the public transportation system. The stops are included into the model as graph nodes. It should be taken into account if besides the decision concerning the structure of the routes, we want to represent the decision concerning whether or not to stop at each bus stop located in the street where the line passes. Observe that this decision directly determines the points where the users can access a given line. In this thesis, we assume that buses stop at every bus stop in their route. Moreover we could want to model the decision concerning the location of the bus stops, although in this thesis we consider a given fixed set of bus

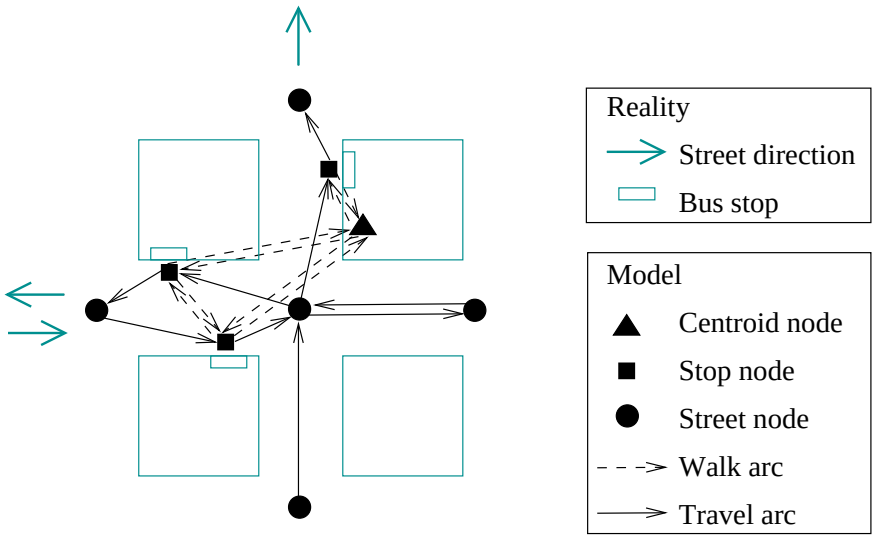


Figure 2.1: Representation of the infrastructure and the demand in terms of graphs

stops. In a general sense, we define a route in terms of graph G as a sequence of adjacent nodes, except those of type centroid.

2.1.3 Simplified model

In order to simplify the models and algorithms developed on this thesis, we adopt a particular structure for the infrastructure graph and the routes defined over it. To represent the infrastructure, we use an undirected graph $G = (V, E)$ whose vertices in V represent the same entities than those represented by the nodes in N of the directed variant (Section 2.1.1). An edge $e \in E$ represents a bidirectional connection between two vertices; its cost c_e is defined analogously to the cost of the arcs in A of the directed variant. Note that this model is in some cases a simplification of the reality, since not all streets have two directions.

The routes are defined as a sequence of adjacent vertices in G, therefore they are composed by undirected edges. We assume that each route has forward and backward directions, having identical on-board travel time. This is also a simplification, since in reality both directions of a given route may be composed by different real streets, therefore their durations may differ slightly (note that lines having significant differences between both directions are not well represented by this model). Circular routes and loops (routes that pass many times by the same vertex) are not allowed.

Increasing the level of detail concerning these hypothesis has different impacts on the difficulty of modeling and solving the problem in the context of this thesis.

2.2 Assignment model

The assignment model determines the way in which the users move themselves from their origins to their destinations, using a given set of public transportation lines. This model

is a critical component of any model of route optimization, since it is necessary to obtain measures concerning the performance of the system, in this case the level of service [33]. Within the scope of this thesis and under the hypothesis stated in Section 1.1, the level of service is measured in terms of the travel time and the level of occupancy of the buses.

The assignment model is a descriptive model which "simulates" the interaction of the users with the buses to obtain the measures of interest. In the context of route optimization, the assignment model is embedded into a normative model which suggests a set of optimal public transportation lines. Such assignment model should apply the hypothesis assumed concerning the behavior of the users with respect to a set of lines; these hypothesis should answer at least the following questions, given an OD pair and a set of public transportation lines:

- Which route or combination of routes will be used by the users for traveling from the origin to the destination?
- What information do they consider for taking that decision?
- Do all users use the same routes?
- How do they behave if there is not sufficient capacity in the routes that they want to use?
- Do all users perceive the same travel time along the same route?

Consider a set of routes $R = \{r_1, \ldots, r_m\}$ defined over G according to Section 2.1.2, with their corresponding frequencies $F = \{f_1, \ldots, f_m\}$, where f_i denotes the number of buses per time unit passing by route r_i. We also refer as *headway* to the inverse of the frequency, i.e., the average time elapsed between two consecutive buses of the line.

We define a directed graph $G^T = (N^T, A^T)$ of trajectories which allows to represent the flows of passengers (hereafter used as synonymous of users) from their origins to their destinations. Each trajectory corresponds to a given OD pair; it is a simple path on G^T that starts on the corresponding origin centroid, ends on the corresponding destination centroid and includes on its extremities, walk arcs connecting these two centroids with stops of the public transportation system (called access and egress arcs). Between these two arcs, the trajectory should include a sequence of arcs that represents the movement of the passengers using the routes in R. A given trajectory may include more that one route if the passenger performs transfers; in this case, we define *stage of travel* as the ordinal of the route in the route sequence followed by the passenger to reach his destination. A trajectory that involves transfers may include walk arcs representing the movement of the passenger between bus stops to take the next line in the sequence.

Observe that different passengers of a same OD pair may use different trajectories (Figure 2.2). The graph G^T should allow to represent all the possible trajectories of all OD pairs, given a set of routes. The structure of the trajectory graph depends on the particular assignment model in which it is used. Figure 2.3 shows two possible structures for the same set of routes shown in (a): the variant (b) does not allow to represent the flows discriminated by line, while variant (c) allows to do that, increasing the size of the model. In [97], different structures of the trajectory graph are presented, depending on the purpose of the assignment model.

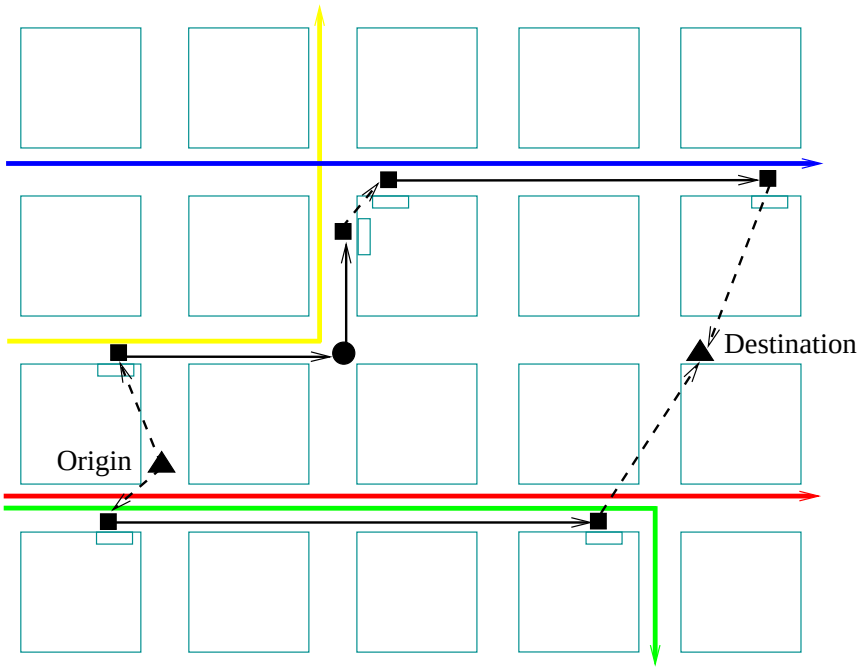
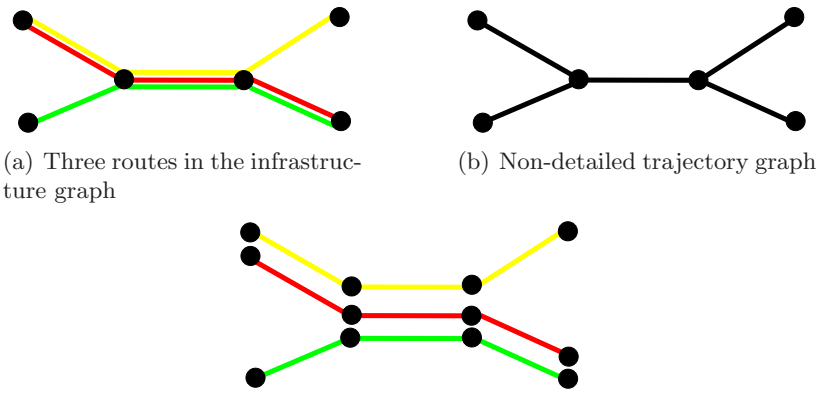


Figure 2.2: Different trajectories for the same OD pair



(a) Three routes in the infrastructure graph

(b) Non-detailed trajectory graph

(c) Detailed trajectory graph

Figure 2.3: Different structures of the trajectory graph

Given a trajectory graph G^T corresponding to a set of routes defined over G and an OD matrix, the assignment model determines the flow v_a of passengers (demand) over each arc $a \in A^T$. The way in which such flows are determined depends on the hypothesis assumed concerning the questions formulated above. The main differences between the assignment models for public transportation existing in the literature, are given by the consideration of the congestion (due to capacities, crowding and passenger boarding/alighting) and the perception of the travel time [97]. In the models that do not consider congestion, given a set of routes R the travel time for a given passenger is fixed and independent on the way in which other users use the lines of R; in the models that consider congestion, the travel time that experience the users depends on the way in which all users use the system [19, 30, 71]. Concerning the perception of the travel time, deterministic models assume that users of the same OD pair perceive the same travel time along a given trajectory; stochastic models assume that such travel time is a random variable [96]. Concerning the level of detail of the information considered by an assignment model, the literature distinguishes between schedule-based models and frequency-based models [97]. The former type requires a complete detail of the timetable of each line and time-dependent demand data. The latter type considers only the frequency of each line and an average value of demand in a given time horizon. While schedule based models allow to obtain detailed output measures (for example, travel times discriminated by time of day), frequency based models obtain averaged output values.

Under hypothesis of no congestion, the assignment model can be posed in terms of a minimization problem; this is possible since an expression that summarizes the optimization of the overall system is consistent with the behavior of each individual of such system. Usually it is assumed that the users choose the lines that they will use by minimizing a generalized cost that includes walking time, waiting time, on-board travel time and the fare; because this last variable is not taken into account in this thesis, the overall travel time (walking + waiting + on-board) is considered as the cost that the users want to minimize when making their decisions. The walking and on-board travel time are attributes of the arcs of the trajectory graph G^T (obtained directly from G), therefore they are fixed. On the other hand, the waiting time depends on the lines that the users consider to perform their trips, therefore they can not be computed a priori. In the context of frequency-based assignment models, the most accepted approaches existing in the literature assume that the arrivals of both passengers and buses to the stops are stochastic processes. Therefore the waiting time of a passenger waiting on a stop for a set of lines $R = \{r_1, \ldots, r_e\}$ with corresponding frequencies $F = \{f_1, \ldots, f_e\}$ is a random variable of mean value [33]

$$E(tw) = \frac{\beta}{\sum_{r_i \in R} f_i}. \tag{2.1}$$

The case $\beta = 1$ corresponds to a negative exponential distribution for the inter-arrival time of buses to the stop (with mean $1/f_i$) and to a uniform distribution of the inter-arrival time of passengers. For deterministic bus arrivals, expression (2.1) with $\beta = 1/2$ corresponds to an approximation of the waiting time [16, 110].

Assuming that the passengers take the first bus arriving to the stop (among the buses

that perform routes of the set R), the probability of using the route r_i is [33]

$$P_i = \frac{f_i}{\sum_{r'_i \in R} f_{i'}}, \tag{2.2}$$

which is known as the *frequency share rule*.

Using expressions (2.1) and (2.2) the assignment problem can be formulated as a problem of minimization of travel time, whose solution is a set of trajectories in G^T; since congestion is not considered, the problem can be separated for each OD pair. The concrete formulation of the problem depends on the hypothesis assumed concerning the way in which the users choose the lines. In the simplest case, for a given OD pair the solution of the assignment problem is a single trajectory; it corresponds to a situation where the passenger selects a priori a single line [34] for which he will wait (this is known as "all or nothing" assignment). Transfers may be considered in this situation, but always restricted to wait for a single line at any stop. An example of this simple trajectory is the one composed by the yellow line and then the blue one, in Figure 2.2. However, in systems based on buses, the most general case includes situations where there is more than one line (eventually overlapped) that can be used to travel between two stops; this implies that the solution to the assignment problem may contain more than one trajectory for the same OD pair (the trajectories consisting in the red line and the green one respectively, in Figure 2.2). The issue posed by this situation was formally presented in [22] as the *common bus lines* problem, later studied and generalized by [29, 110], that consider multiple lines (not necessarily overlapped).

2.3 Transit Network Design Problem

In this section we define the elements of the problem that we take into account in the thesis. Given that in different instances of this research we have studied concrete variants or aspects of the problem, approached with different techniques, the concrete terminology, definitions and notation are given in the corresponding parts of the document.

In general terms, we included the following elements of the problem in our research concerning models and algorithms for route optimization in public transportation systems:

- The interest of the users.
- The interest of the operators.
- The behavior of the users.
- Transfers and demand covering constraints.
- Constraints regarding infrastructure and bus capacity.

In the following we discuss in general terms how these elements are modeled.

The models of route optimization have as main objective to design a system with the highest possible level of service to the users [33], measured in terms of the travel time and the occupancy levels of the buses. Given a set of routes defined over graph G, these

measures are obtained from the flows v_a determined by the assignment model, over arcs a of the corresponding trajectory graph G^T.

The interest of the operators is usually the maximization of profit, which under the hypothesis of inelastic demand (therefore fixed income, assuming that the whole demand is covered by routes) is equivalent to the minimization of cost. The real operator's cost is difficult to express in a general sense [109], given that it depends on each particular case (fare structure, existence of subsidies). For that reason, the size of the fleet is commonly used as a proxy for operator's cost, defined as the number of buses necessary to operate a set of routes $R = \{r_1, \ldots, r_m\}$ with corresponding frequencies $F = \{f_1, \ldots, f_m\}$. The number of buses to operate route r_i is calculated as $f_i \sum_{a \in r_i} c_a$, assuming that the route is a cycle (for example, with forward and backward directions connected by their ends). Observe that the fleet size is an important component of the operator's cost, since it is directly proportional to the route length and the staff (drivers) time needed to operate the buses.

The behavior of the users with respect to a set of lines is considered by the assignment model. In the route optimization models studied in this thesis, we consider assignment models that take into account the minimization of travel time (including waiting time), transfers and multiple lines (Section 2.2).

Both objectives of users and operators should be taken into account by the regulator when planning the routes; in terms of the optimization model, its decisions are the values given to the variables that represent the routes (defined in terms of G) and frequencies. On the other hand, the behavior of the users (represented by the assignment model) is reflected by flows v_a over arcs a of the trajectory graph G^T, for a given set of routes and frequencies.

Once an OD matrix is available to represent the needs of public transportation lines, we may consider to cover all the required demand. We say that the demand is covered by a set of routes R if for each OD pair (i, j) there is at least one trajectory in G^T connecting nodes i and j, which may or not include transfers. In most real systems, usually it is impossible to connect every OD pair directly (without transfers) because the resulting system has a high number of lines (therefore the operation cost is high). Transfers can be included as a penalized term into the objective function of the model of route optimization, however this approach does not allow the planner to control the amount of transfers on the optimal solution. An alternative is to include demand covering constraints, introduced by [9]. These constraints state that for any feasible solution, at least a given percentage of the total demand $\sum_{i,j=1..n} d_{ij}$ should be covered with no more than a given number of transfers. For example: at least the 70% of the whole demand should be covered without transfers (direct trips), and the remaining 30% with one transfer at most. Related to this constraints, we should take into account that a trajectory includes walk arcs that connect zone centroids (where the demand is generated) with public transportation stops (where the lines pass); therefore the zoning of a city and its corresponding coding in the graph G should include walk arcs of reasonable distance, otherwise the demand can not be considered as covered.

The capacity of the infrastructure can be represented through the capacity of the streets over which routes are defined. In terms of the graph G, this capacity can be included as an attribute of street arcs. It should express the maximum number of buses that can pass over

each arc by time unit; we note that no congestion effect is considered before the capacity is reached. This constrains the way in which routes and frequencies are defined; it affects the decisions of the planner. The bus capacity constraint expresses a maximum allowable flow on each line l; it links the flows v_a determined by the assignment model with the capacity of the line, defined as $f_l\omega$ (where f_l is the frequency of line l and ω is the capacity of each bus, expressed in number of persons). Although the bus capacity constraint seems to be expressed in a similar way as the street capacity constraint, it impacts very differently when it is included in the route optimization model, since it affects the behavior of the users (represented by the flows) in response to decisions of the planner (represented by routes and frequencies). The models studied in this thesis consider the bus capacity constraint so as to ensure sufficient capacity in such a way that the users can use the routes that they desire. In other words, the users will perceive lines of unlimited capacity; but that perception should be ensured by the optimization model, otherwise the assignment sub-model will produce flows that are not consistent with the hypothesis stated in Section 2.2 (among them: no congestion). The same modeling approach is adopted in [9, 73]. However there are real cases where it is technically impossible to ensure sufficient capacity in the routes that the users desire to use; despite the existence of public transportation modes with high frequency possibilities (BRT, underground trains), the frequencies can not be increased arbitrarily above the physical limits (operational constraints related to safety also come into play). In these cases, the behavior of the users under hypothesis of congestion should be modeled, i.e, we should model the situation of users who are forced to wait for the next bus due to lack of line capacity; under this scenario the user may choose other set of lines, different from the one that would be chosen in absence of congestion. This issue entails to solve an equilibrium problem [108] within the assignment model [19, 30, 55]. This characteristic adds complexity in a high degree to the problem of route optimization; the literature published in this specific topic is very scarce [47].

With the purpose of using an abbreviated name for the problem of route optimization in public transportation systems defined in general terms on this section, we use the denomination Transit Network Design Problem introduced by [9]. We refer as solution to this problem, a set of routes (as defined in Sections 2.1.2 and 2.1.3) with their corresponding frequencies. The term "network" comes from the fact that the users perceive a set of routes as a network, obtained by merging all the routes into a single entity. The term "transit network" usually refers to public transportation systems based on buses, which are the subject of this thesis. Other terms have been used to denominate the problem of route optimization, as the "line planning problem", which also includes systems based on trains [106].

Chapter 3

Mathematical programming formulation*

In this chapter we present a mathematical programming formulation for the TNDP. We propose an explicit formulation that models the aspects of the problem identified in Section 2.3. The issues that motivate this part of the thesis are the following ones:

- The research concerning this specific topic is relatively recent and scarce.
- Despite the existence of algorithms based on explicit mathematical programming formulations that have solved real instances of the problem, the component related to the assignment sub-model is not realistic for systems based on buses (specially under hypothesis 6 of Section 1.1). The modeling of the waiting time of the users and the assignment to multiple routes are two key aspects of such systems that are not included in the existing formulations.
- The influence of demand covering and bus capacity constraints in the structure of the resulting formulation has not been studied.
- We are interested in obtaining lower bounds or (when possible) optimal solutions. These results can be useful to evaluate heuristic approaches to the problem. Moreover we want to explore the applicability of the formulation to solve real instances of the problem.

A specific literature review is included at the beginning of the chapter, followed by an analysis that justifies partially the motivations for this part of the thesis. We propose a base formulation that minimizes the total travel time of the users, subject to a constraint on the fleet size. The assignment model known as "optimal strategies" [110] is used to model the behavior of the users. The resulting formulation is mixed integer nonlinear, with a super-polynomial number of variables that represent the routes. The existence of these variables is the main difficulty to solve exactly the problem, due to its discrete nature and high number. The frequencies are represented as real variables. The nonlinear nature of the formulation is due to the constraint that splits the flow of passengers among

* Part of the content of this chapter was published in [82] and presented in [83].

different bus lines; we propose a linearization of this constraint, casting the formulation into a MILP one. We then add the demand covering constraints, which increase the size of the model considerably. The constraint related to street capacity is directly incorporated into the model, while including the bus capacity constraint leads to a bilevel programming formulation [12]. We then present some alternatives to solve that formulation, based on existing techniques for this kind of problem. Finally, we present computational results applying the linearized formulation without demand covering, infrastructure and bus capacity constraints. We use very small test instances, either specially created for this work or obtained from the literature, for which optimal solutions are obtained. The case of the city of Rivera (Appendix A) is used to show that the formulation could be employed to solve a particular application of the TNDP.

3.1 Literature review

The existing mathematical programming formulations for the TNDP are relatively recent and scarce. This is possibly due to the difficulties to formulate the problem [20] and to devise efficient methods to solve it. In the following we review each one of the existing formulations.

In [120], a MILP formulation that minimizes the operator's cost subject to a bus capacity constraint is proposed. A salient characteristic of this formulation is that it allows to generate implicitly the structure of the routes; however, this requires that a maximum number of routes in the solution should be specified. The routes are represented by introducing node labels that indicate whether a given node belongs to a given route and its ordinal in the route. The behavior of the users is not modeled; the demand is assigned to the routes so as to satisfy the bus capacity; this is not a realistic assumption in our context. Despite having a high number of variables, the total size of the model (number of variables and constraints) is of polynomial order with respect to the size of the problem.

In [58], the authors propose a MILP formulation that minimizes an objective function including line cost, number of transfers and on-board travel time; therefore the interest of the operators and both interest and behavior of the users are considered simultaneously. These different terms are weighted by coefficients that express the relative importance of each one. Frequencies are not decision variables and the waiting time is ignored in the behavior of the users. The constraint set includes street capacity, line length and maximum number of transfers. The model assumes that a pool of candidate lines is provided and even the possible trajectories that users will use are identified beforehand. Since neither bus capacity is considered nor the frequency share rule is applied, all the demand corresponding to the same OD pair uses the same trajectory.

The TNDP is formulated in [107] as an extended multi-commodity flow problem that minimizes the on-board travel time and the number of transfers, subject to a budgetary constraint (fixed cost of each line) and the bus capacity constraint. The formulation is a MILP one. The transfers are modeled by using a particular graph structure called *change and go network*. That graph includes one arc for each route that shares the same street arc with other routes and one arc for each possible transfer between routes that share the same bus stop in the system. Given that the objective function minimizes the travel

time, both the interest of the users and its behavior are modeled by the same component of the formulation. Implicitly it is assumed that the users ignore the waiting time when choosing the lines. The implicit assignment model is constrained by the bus capacity, thus the formulation models a situation of capacitated user equilibrium [27]. The size of the formulation is super-polynomial.

In [15], a MILP multi-commodity flow formulation is proposed. It minimizes the travel time of the users, and fixed and variable costs of the operators, under street and bus capacity constraints. As in [107], this formulation implicitly assumes that the users ignore the waiting time and take into account the bus capacity for choosing the lines. Transfers are ignored, meaning that the users may need to perform an unbounded number of transfers in the optimal solution. While [107] models the interest of the operator as a budgetary constraint, in this work it is added as a term in the objective function, resulting in a multi-objective optimization model. The size of the formulation is also super-polynomial.

Table 3.1 presents a summary of the formulations mentioned above, showing the way in which the relevant aspects of the problem identified in Section 2.3 are approached. The formulation [120] models the lowest number of those aspects, therefore it is somehow the least realistic one; however it has the capacity of generating implicitly the routes. The formulations of [15, 107] are solved by applying decomposition techniques, where the routes are generated by solving the corresponding associated subproblem.

By analyzing the formulations reviewed in this section, we can identify the aspects of the TNDP which are more difficult to model:

- *The structure of the routes.* In the four formulations, the routes are represented by sequences of adjacent edges in an undirected graph. These routes are generated in [15, 107] by solving the subproblem corresponding to the decomposition technique used to solve the problem. In [120] the routes are generated implicitly through variables specially introduced; in particular a non-trivial problem is to exclude cyclic lines from the solution. Note that in Section 2.3 we have assumed that routes defined over the directed infrastructure graph are cyclic; those cycles can be obtained by merging both forward and backward directions of the line. However, where an undirected graph is used to represent the infrastructure, the cycle is implicit in a route defined as a simple path. In [120], cycles in undirected routes are not allowed.

- *Transfers.* When using a flow formulation, if the flow is assigned ignoring the individual routes (simply each edge of the graph is enabled if at least one route passes over it), the transfers will be allowed but unbounded. We say that the transfers are unbounded if in the optimal solution, there may be OD pairs that need to perform a high number of transfers to reach the destination; this is the case of the formulation proposed in [15]. On the other hand, the formulation proposed in [107] is based on the *change and go network* structure, that allows to represent individually each route and to identify transfers in the flow assignment; in this model, the transfers are considered either in the behavior of users as on its interest, through the objective function. In [58] both lines and passenger trajectories are identified beforehand; this is only possible when using small-sized instances of the problem. By doing this, transfers can be identified in the formulation as the number of lines used by each OD pair and they also can be counted in the objective function and/or bounded in

	Interest of the users	Interest of the operators	Behavior of the users	Bus capacity	Transfers
Wan and Lo [120]	Not modeled (only minimum frequency).	Min. variable operation cost.	Not modeled.	Included as constraint.	Not allowed.
Guan et al. [58]	Min. on-board travel time and transfers.	Min. fixed line cost.	Min. on-board travel time and transfers.	Not modeled.	Allowed and bounded; counted in the objective function.
Schöbel and Scholl [107]	Min. on-board travel time and transfers.	Constraint on fixed operation cost.	Min. on-board travel time and transfers. User equilibrium according to bus capacity constraint.	Included as constraint.	Allowed and unbounded; counted in the objective function.
Borndörfer et al. [15]	Min. on-board travel time.	Min. fixed and variable operation costs.	Min. on-board travel time. User equilibrium according to bus capacity constraint.	Included as constraint.	Allowed and unbounded.

Table 3.1: Mathematical programming formulations for the TNDP

a constraint.

- *The behavior of the users, in particular the waiting time.* This aspect is not taken into account in any of the presented formulations. The difficulty in modeling the waiting time lies on its nonlinear dependency with respect to the frequencies (expression (2.1)). The modeling of the assignment to multiple routes (when included) adds complexity to the formulation.

Although the solution methods based on mathematical programming have solved instances of the problem of moderate size ([15] apply their method to a real city comprising 27 bus lines), for systems based on buses we consider that the waiting time should be taken into account in order to have a realistic modeling. Also it is desirable that the transfers can be included as a constraint in the optimization model. The existing formulations are suitable under scenarios where the users are not sensitive to the waiting time and transfers, due either to high frequency services, user information systems (published timetables, real time information) or coordinated transfers.

3.2 Base formulation

The proposed base formulation considers each OD pair as a commodity that flows through the trajectory graph defined by the infrastructure graph and the routes defined over it; we adopt the simplified model described in Section 2.1.3. The interest of the users is represented by the minimization of an objective function that includes the total travel time. The interest of the operators is formulated by a maximum fleet size constraint. The behavior of the users is represented by the "optimal strategies" assignment model [110]. In the particular notation used to express the formulation, the origin-destination demand is given by a set of commodities K such that for each OD pair (i, j), there is a corresponding $k \in K$ with associated values $O_k = i$, $D_k = j$ and $\delta_k = d_{ij}$.

3.2.1 Assignment sub-model

In this section we explain the assignment sub-model used to represent the behavior of the users. We adopt the model called "optimal strategies"; the main concepts and formulations presented on this section are taken from its original publication [110].

We adopt this model because it has the following desirable characteristics related to the definition of the problem studied in this thesis (Section 2.3):

- It makes assumptions about the behavior of the users that are consistent with those considered in this work; in particular, the model assumes that the users want to minimize their travel time (walking+waiting+on-board).
- It models the problem of assignment to multiple routes.
- It has an explicit and compact mathematical formulation.
- It is a frequency-based assignment model, suitable to be included into an optimization model that handles information at the same level of aggregation (the decision variables are routes and frequencies).

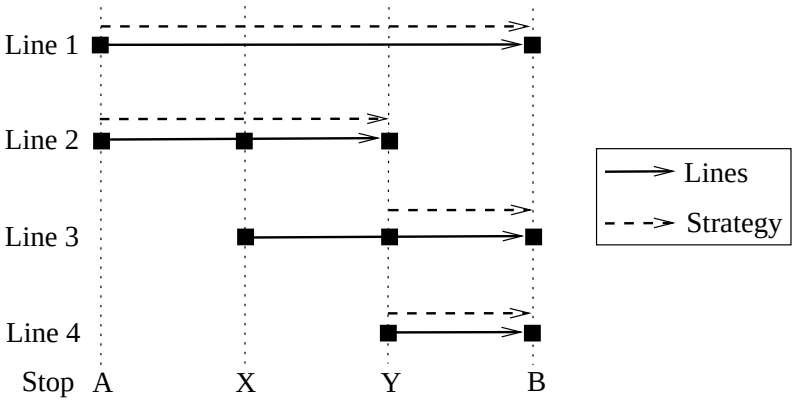


Figure 3.1: Example of a strategy to travel from A to B

The model is based on the concept of *strategy*, which is a set of rules that when applied, allow the user to reach his destination. An example of strategy for passengers traveling from A to B in Figure 3.1 is the following:

"At A, take the following bus of lines 1 or 2; if line 1 was taken, finish at B; if line 2 was taken, transfer at Y to line 3 or 4 and finish at B."

According to his strategy, the passenger will travel from his origin centroid to his destination centroid, passing by the following stages: (1) walk to the origin stop, (2) wait at the stop, (3) travel on-board the bus and (4) walk from the destination stop. Eventually steps 2 and 3 may be repeated in case of performing transfers, that also may imply to walk between stops.

A strategy for a given OD pair $k \in K$ can be defined as a set of possible trajectories determined a priori, defined over the graph G^T; each trajectory is a simple path from O_k to D_k. Note that if we restrict a strategy to a single trajectory, we are modeling a situation that is not real in the context of the assignment to multiple lines; in the example of Figure 3.1, the strategy is composed by three trajectories. The model assumes that a given user selects the strategy that minimizes his total travel time. To do this, he will select a priori (i.e., before leaving the place where the trip is originated, in this case the origin centroid) a set of "attractive" lines among all the possible lines that connect the bus stops located near their origin and destination centroids. In this process, the passenger considers information related to the on-board travel time of all the lines of the system (given by the cost of the travel arcs in G^T) and the walking times in all the system (given by the cost of the walk arcs in G^T); he also knows the frequency of all the lines of the system, needed to compute the waiting time. While waiting at the bus stop, the user will take the first bus passing by that stop, belonging to the set of attractive lines determined a priori. A strategy is optimal if it minimizes the total travel time. Since the model of optimal strategies is probabilistic, in the sense of the assignment of demand corresponding to the same OD pair to multiple routes, the measure that is minimized is the total expected travel time. It is worth mentioning that this model assumes that the user is perfectly informed and even is able to manage all that information in order to

determine the optimal strategy (which eventually may be quite complex).

For simplicity, in the following we will assume that all vertices of the infrastructure graph are of type street, stop and centroid at the same time; this implies that the demand can be generated at any vertex and walk arcs are not considered. The computation of the optimal strategy is stated as an optimization problem defined over a particular structure of the trajectory graph. It includes nodes and arcs specific for each route and arcs that represent wait. More precisely, given an infrastructure graph G and a set of routes R (with corresponding frequencies) defined over G according to the simplified model stated in Section 2.1.3, the arcs in A^T of the trajectory graph can be:

- Travel arcs A^V, that model the movement of a bus performing a route of R (and the passengers that travel on-board), from a given vertex to other one of G.
- Wait arcs A^W, that model a passenger waiting for a given line of R, in a vertex of G.
- Destination arcs A^D, that model the end of the travel.

The sets of arcs mentioned above are such that $A^T = A^V \cup A^W \cup A^D$; these sets are disjoint.

The set of nodes N^T of the trajectory graph is obtained as follows. For each route $r \in R$ that passes by vertex $v \in V$, a node $n_{rv} \in N^T$ is generated; then for each route r that passes by edge $e = [i, j] \in E$, forward and backward travel arcs $\overrightarrow{a}_{re} = (n_{ri}, n_{rj})$ and $\overleftarrow{a}_{re} = (n_{rj}, n_{ri})$ respectively are generated, whose costs are such that $c_e = c_{\overrightarrow{a}_{re}} = c_{\overleftarrow{a}_{re}}$. For each OD pair $k \in K$, their corresponding origin and destination nodes $O_k^N, D_k^N \in N^T$ are generated, as well as a wait arc (O_k^N, n_{rO_k}) for each route r that passes by O_k in G (also called origin arcs). Destination arcs are generated analogously. Wait and destination arcs have zero cost. The value f_a refers to the frequency of the route from which the arc $a \in A$ was generated. An example of the relation between the infrastructure graph and its corresponding trajectory graph for a given set of routes, is shown in Figure 3.2.

The model of optimal strategies is formulated by expressions (3.1)-(3.5), that we call ASIG1. In order to simplify the notation, hereafter we omit the superindexes T of the sets N and A of the trajectory graph and we assume a single OD pair (therefore we omit the subindexes k of O, D and δ). We define values b_n for $n \in N$, such that $b_O = \delta$ (demand on the origin node), $b_D = -\delta$ (demand on the destination node) and $b_n = 0$ otherwise. The sets $A_n^+, A_n^- \subset A$ denote outgoing and incoming arcs respectively from (to) node n. The variable V_n represents the demand quantity that flows through node $n \in N$ and the binary variable x_a states if arc $a \in A$ is part of the optimal strategy. The variable v_a (already defined at Section 2.2) represents the flow assigned to arc $a \in A$.

The objective function states the minimization of the on-board travel time (first term) plus the expected value of the waiting time (second term, using expression (2.1) with $\beta = 1$). Constraint (3.2) states the split of demand flow at a given node among its outgoing arcs, using expression (2.2), while constraint (3.3) states the flow conservation at nodes. Note that the formulation states that passengers wait at every node of the graph, therefore we assume that arcs that are not of wait type have a very high frequency (consequently, the corresponding waiting time can be ignored).

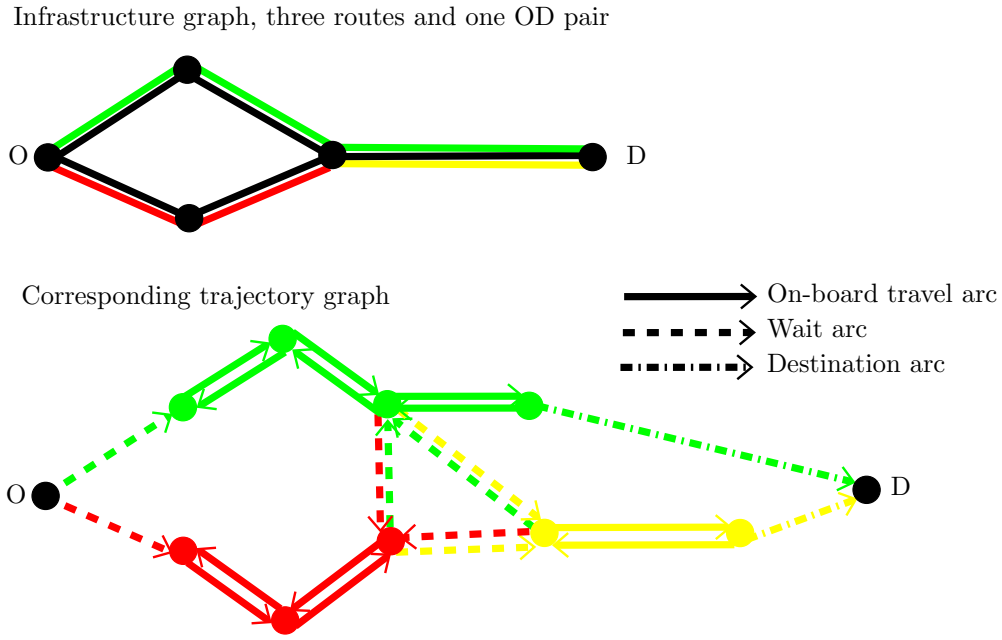


Figure 3.2: Trajectory graph corresponding to a given infrastructure graph and set of routes

$$\min_{v,V,x} \quad \sum_{a\in A} c_a v_a + \sum_{n\in N} \frac{V_n}{\sum_{a\in A_n^+} f_a x_a} \tag{3.1}$$

s.t.

$$v_a = \frac{f_a x_a}{\sum_{a'\in A_n^+} f_{a'} x_{a'}} V_n \qquad \forall\ a \in A_n^+, n \in N, \tag{3.2}$$

$$V_n = \sum_{a\in A_n^-} v_a + b_n \qquad \forall\ n \in N, \tag{3.3}$$

$$V_n \geq 0 \qquad \forall\ n \in N, \tag{3.4}$$

$$x_a \in \{0,1\} \qquad \forall\ a \in A. \tag{3.5}$$

We can observe that ASIG1 is not linear (expressions (3.1) and (3.2)) with mixed variables (v and V are real while x is binary). However, by performing the following change of variables:

$$w_n = \frac{V_n}{\sum_{a\in A_n^+} f_a x_a} \quad \forall n \in N, \tag{3.6}$$

substituting the constraint of nonnegativity of flow at nodes ($V_n \geq 0$) by its analog for arcs ($v_a \geq 0$) and applying properties of the feasible set determined by the resulting constraints, the authors [110] obtain the formulation given by expressions (3.7)-(3.11), called ASIG2. In this formulation, A_n^{W+} denotes the set of outgoing wait arcs from node n. Note that this formulation, which is linear with real variables, has a strong resemblance with the formulation of the minimum cost path problem [2]. The differences consist in

that ASIG2 includes the term that represents the waiting time in the objective function (3.7) and the constraint (3.9) that splits the flow. This constraint causes that the solution to the problem is not a single trajectory, instead it is a set of trajectories that represent the different (attractive) lines that the user considers as part of his strategy; this concept is denominated as *hyperpath* in [95].

$$\min_{v,w} \quad \sum_{a\in A} c_a v_a + \sum_{n\in N} w_n \tag{3.7}$$

$$\text{s.t.} \quad \sum_{a\in A_n^+} v_a - \sum_{a\in A_n^-} v_a = b_n \qquad \forall\ n \in N, \tag{3.8}$$

$$v_a \le f_a w_n \qquad \forall\ a \in A_n^{W+}, n \in N, \tag{3.9}$$

$$v_a \ge 0 \qquad \forall\ a \in A, \tag{3.10}$$

$$w_n \ge 0 \qquad \forall\ n \in N. \tag{3.11}$$

So far, we have excluded the access/egress arcs and centroids from our presentation. We note that these elements can be easily incorporated to this model. Centroids can be linked through walking arcs (which are independent of the routes) to origin and destination nodes that represent origins and destination stops in the model described above. Formulations ASIG1 and ASIG2 will be still valid. On the other hand, walk arcs between bus stops (which may be used when performing transfers) can not be added directly to the model illustrated by Figure 3.2; note that the model assumes that transfers take place at the same bus stop.

3.2.2 Formulation of the model of route optimization

In the following we present the mathematical programming formulation proposed for the TNDP. Let R be the set of all possible routes defined over G according to the simplified model stated in Section 2.1.3. We define the binary variable x_r that takes value 1 if route $r \in R$ is included in the solution, 0 otherwise, while the real valued variable f_r indicates the frequency of route $r \in R$ expressed in buses per time unit. Now we add the subindex k to variables v and w and to the constant value b, to indicate their corresponding OD pair $k \in K$. We also use the following notation: $E_r \subseteq E$ are the edges of the route $r \in R$, $r(a) \in R$ is the route that originated the arc $a \in A$ (each arc belongs to exactly one route) and B is a maximum allowed fleet size.

The formulation called OPT1 is given by expressions (3.12)-(3.20). Its objective function (3.12) represents the interest of the users and their behavior (along with constraints (3.14) and (3.15), which as the objective function, are part of the assignment sub-model). Constraint (3.13) represents the interest of the operator while constraint (3.16) states that the users only can use routes that are part of the solution. Formulation OPT1 is nonlinear (constraint (3.15)) and mixed integer (variable x is binary).

We note that the only constraint imposed on the frequencies is the nonnegativity constraint (3.17). This may be unrealistic, since in real systems both lower and upper levels on frequencies should be imposed; the former is to ensure a minimum level of service in terms of waiting time, while the latter is for technical reasons. Solutions to

OPT1 will minimize the total travel time (which includes the waiting time) subject to a maximum fleet size; these quantities are indirectly and directly proportional to the frequencies, respectively. Thus, the optimal solution to this problem may have arbitrary low or high values of frequencies; this issue will be addressed in Section 3.2.3.

$$\min_{x,f,v,w} \quad \sum_{k\in K}(\sum_{a\in A} c_a v_{ak} + \sum_{n\in N} w_{nk}) \tag{3.12}$$

$$\text{s.t.} \quad \sum_{r\in R} 2f_r \sum_{e\in E_r} c_e \leq B, \tag{3.13}$$

$$\sum_{a\in A_n^+} v_{ak} - \sum_{a\in A_n^-} v_{ak} = b_{nk} \qquad \forall\, n \in N, k \in K, \tag{3.14}$$

$$v_{ak} \leq f_{r(a)} w_{nk} \qquad \forall\, a \in A_n^{W+}, n \in N, k \in K, \tag{3.15}$$

$$v_{ak} \leq \delta_k x_{r(a)} \qquad \forall\, a \in A, k \in K, \tag{3.16}$$

$$f_r \geq 0 \qquad \forall\, r \in R, \tag{3.17}$$

$$v_{ak} \geq 0 \qquad \forall\, a \in A, k \in K, \tag{3.18}$$

$$w_{nk} \geq 0 \qquad \forall\, n \in N, k \in K, \tag{3.19}$$

$$x_r \in \{0,1\} \qquad \forall\, r \in R. \tag{3.20}$$

3.2.3 Linearization

The only nonlinear expression in OPT1 is the constraint of flow splitting (3.15). Note that in the original assignment model [110], this expression is linear since the frequencies are problem data; on the other hand, frequencies are decision variables in the model of route optimization. We propose a linearization of this constraint, by discretizing the domain of frequencies. A new parameter of the problem is introduced, that represents all the possible values of frequencies on routes:

$$\Theta = \{\theta_1, \ldots, \theta_p\}, \tag{3.21}$$

indexed by f.

Then, in OPT1 the real variable f_r should be substituted by the new binary variable y_{rf} which takes value 1 if route r has frequency f, 0 otherwise. The expressions that involve variable f in OPT1 should be rewritten in terms of y. The linearized formulation called OPT2 is given by expressions (3.22)-(3.32), where $f(a)$ refers to the index in Θ of the frequency corresponding to the wait arc $a \in A^W$ (which is a fixed value). The trajectory graph should include a wait arc for each element in Θ (Figure 3.3 shows an example for $|\Theta| = 3$); in that graph, only the arcs corresponding to the frequency selected for each route will be enabled (constraint (3.27)).

Observe that the constraint of flow splitting (3.25) is linear in OPT2, given that $\theta_{f(a)}$ is a constant value. Therefore we obtained a MILP formulation. It is worth mentioning that the linearization increases the size of the model, since a new binary variable y is added, as well as a number of wait arcs proportional to the size of the set Θ.

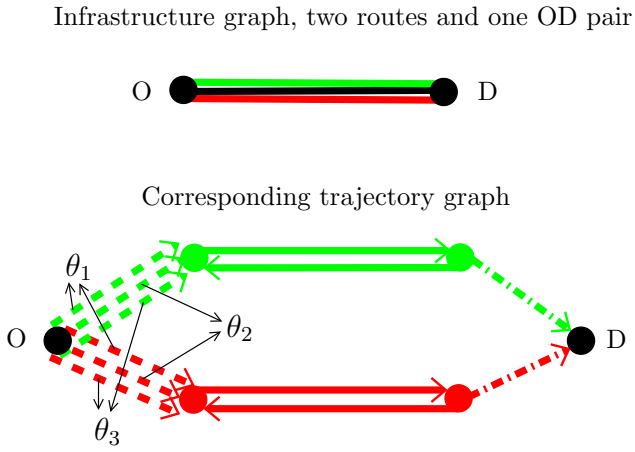


Figure 3.3: Trajectory graph for the linearized formulation

$$\min_{x,y,v,w} \quad \sum_{k\in K}(\sum_{a\in A} c_a v_{ak} + \sum_{n\in N} w_{nk}) \tag{3.22}$$

$$\text{s.t.} \quad \sum_{r\in R} 2 \sum_{f\in\Theta} \theta_f y_{rf} \sum_{e\in E_r} c_e \leq B, \tag{3.23}$$

$$\sum_{a\in A_n^+} v_{ak} - \sum_{a\in A_n^-} v_{ak} = b_{nk} \qquad \forall\, n \in N, k \in K, \tag{3.24}$$

$$v_{ak} \leq \theta_{f(a)} w_{nk} \qquad \forall\, a \in A_n^{W+}, n \in N, k \in K, \tag{3.25}$$

$$v_{ak} \leq \delta_k x_{r(a)} \qquad \forall\, a \in A, k \in K, \tag{3.26}$$

$$v_{ak} \leq \delta_k y_{r(a)f(a)} \qquad \forall\, a \in A^W, k \in K, \tag{3.27}$$

$$\sum_{f\in\Theta} y_{rf} = x_r \qquad \forall\, r \in R, \tag{3.28}$$

$$v_{ak} \geq 0 \qquad \forall\, a \in A, k \in K, \tag{3.29}$$

$$w_{nk} \geq 0 \qquad \forall\, n \in N, k \in K, \tag{3.30}$$

$$x_r \in \{0,1\} \qquad \forall\, r \in R, \tag{3.31}$$

$$y_{rf} \in \{0,1\} \qquad \forall\, r \in R, f \in \Theta. \tag{3.32}$$

3.3 Enforcing transfer, infrastructure and bus capacity constraints

In this section we incorporate to the base formulation developed in Section 3.2, constraints related to transfers, infrastructure and bus capacity. These elements of the TNDP were identified in Section 2.3 as important aspects to be included into a realistic model for a public transportation system based on buses under the hypothesis stated in Section 1.1. The transfer constraints are written by means of an auxiliary structure of the trajectory

graph. The street capacity constraint is directly incorporated to the model, while the bus capacity constraint (jointly with the transfer constraint) leads to a discussion related to a bilevel mathematical programming formulation for the TNDP.

3.3.1 Transfer constraints

The modeling of transfers in OPT2 depends on the structure of the trajectory graph used. The definition of G^T given in Section 3.2.1 includes transfer arcs, as illustrated in Figure 3.2. It results in a structure that is very similar to the *change and go network* used in [107]. By using this structure, we can obtain an optimal solution where an unbounded percentage of the total demand $\sum_{k \in K} \delta_k$ performs an unbounded number of transfers. This could be avoided by setting a high cost c_a to each arc a that represents a transfer, however it is not clear the magnitude that should have that cost to obtain a desired result. On the other hand, if we eliminate the transfer arcs from G^T, there can exist an OD pair that is forced to travel using a single route with a high travel time, which could be decreased by performing transfers. An alternative is to include demand covering constraints (Section 2.3). These constraints were introduced in [9] but they have not been yet included into an explicit mathematical formulation, instead they are verified algorithmically in the context of heuristics [9, 101]. In their original definition, demand covering constraints accept solutions where a certain portion of the total demand may be unsatisfied, i.e. it can not be covered by performing any number of transfers. In our models, we are restricted to the case that the whole demand should be covered. For this reason, we call *transfer constraints* to our particular version of demand covering constraints.

To model the transfers and to have control over them in the optimization model, we introduce a parameter that states the maximum allowed number of transfers, given by the positive integer constant τ. Then, we generate $\tau + 1$ trajectory graphs $G_i^T, i \in [1..\tau + 1]$ corresponding to each possible stage of travel of the passengers. The origin arcs connect the origin nodes only with the graph corresponding to the first stage, G_1^T; destination arcs connect the destination nodes with the graphs of all stages. The arcs that represent transfers have the form $(n_{r_1 v}, n_{r_2 v})$, where $r_1, r_2 \in R$ are routes passing by the same vertex v, the first node belongs to the graph G_i^T and the second node belongs to G_{i+1}^T, for all $i \in [1..\tau]$. Then, the sum of the flows of destination arcs connected to graph G_i^T will represent the number of users that make $i - 1$ transfers. Figure 3.4 shows an example of such a graph structure for the scenario of Figure 3.2, imposing $\tau = 1$ (one transfer at most). We note that the size of the model increases considerably, since the graph structure is replicated for each stage of travel and new transfer arcs are added.

Then, the transfer constraints can be written as

$$\sum_{k \in K} \sum_{a \in \cup_{i=1..s} A_i^D} v_{ak} / \sum_{k \in K} \delta_k \geq \Delta^s \quad \forall\ s \in [1..\tau], \tag{3.33}$$

where $A_i^D \subset A$ denotes the set of destination arcs of G_i^T, and $0 \leq \Delta^1 \ldots \leq \Delta^\tau \leq 1$ are real values such that Δ^s is the proportion of the total demand $\sum_{k \in K} \delta_k$ that should be covered with no more than $s - 1$ transfers (i.e., with at most s stages of travel).

By adding constraint (3.33) to formulation OPT2 and using the structure of the trajectory graph proposed above, we can control the transfers as stated in Section 2.3. However,

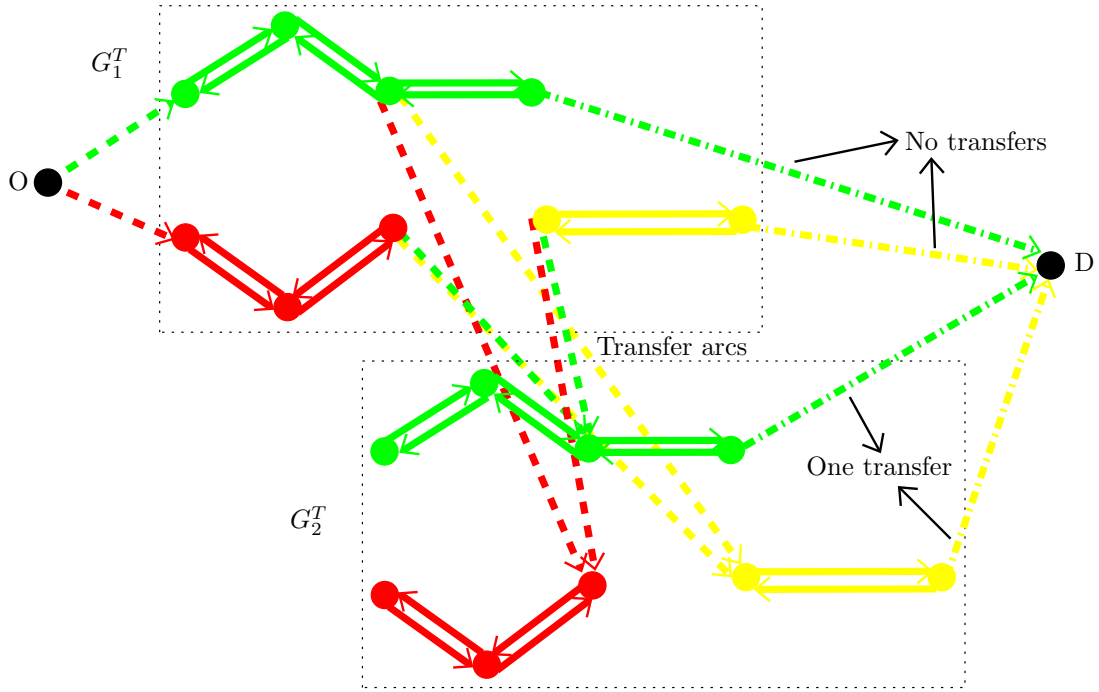


Figure 3.4: Trajectory graph used to express transfer constraints

by doing this we can obtain solutions that are not consistent with the behavior of the users stated by the assignment model of optimal strategies. We resume this issue in Section 3.4, related to a bilevel formulation for the TNDP. We note that alternatively, expression (3.33) could be added to the objective function as a penalized term. However, this approach requires determining the value of such penalty and even that value is assumed to be the same for both planner and user.

3.3.2 Street and bus capacity constraints

We consider to incorporate infrastructure and bus capacity constraints to OPT2. In our context, the infrastructure is given by the streets over which the routes will be defined.

Street capacity constraint

The capacity of the streets is given for each edge $e \in E$ by a positive real value κ_e which expresses the maximum number of buses that can pass by e per time unit. Observe that since the infrastructure graph is undirected, the capacity refers to the total number of buses passing in both directions of the edge. The street capacity constraint relates the edge capacity with the frequency of each line that passes by it, as follows:

$$\sum_{r \in R} \sum_{f \in \Theta} y_{rf} \theta_f \Lambda_{er} \leq \kappa_e \quad \forall\, e \in E, \tag{3.34}$$

where Λ_{er} is a binary constant that takes value 1 if route r passes by edge e, 0 otherwise.

Note that by adding this constraint we are not affecting variables that model the behavior of the users, so it can be added directly to formulation OPT2.

Bus capacity constraint

The capacity of the bus represents the maximum number of passengers that can travel inside the vehicle. The bus capacity constraint relates the flow of demand assigned to travel arcs of the trajectory graph, with the capacity of the lines, as follows:

$$\sum_{k \in K} v_{ak} \leq \sum_{f \in \Theta} y_{r(a)f} \theta_f \omega \quad \forall\ a \in A^V, \tag{3.35}$$

where ω denotes the capacity of the bus. This parameter can be given through a single value that expresses the total capacity of the bus or through the product of the capacity of seated passengers and a coefficient (greater than 1) that states the allowed load factor (standing passengers over the seat capacity of the bus).

It is worth mentioning that although the expression of the bus capacity constraint is similar to the one corresponding to the street capacity constraint, it impacts in a very different way in formulation OPT2. Since the bus capacity constraint is written in terms of variables that represent decisions of passengers, adding it directly to OPT2 may result in solutions that are not consistent with the assignment model of optimal strategies. We resume this issue in Section 3.4.

3.4 Bilevel mathematical programming formulation

Formulation OPT2 models decisions taken by different actors in the context of the TNDP. Variables x and y represent decisions of the planner who decides which lines and frequencies to establish, while variables v and w represent decisions of the users who decide which lines to use, among those determined by the planner. The objective function of OPT2 models at the same time, the interest of the users (taken into account by the planner, when determining the lines and frequencies) and their behavior (determined by themselves, by choosing the lines according to the hypothesis of the assignment model). Therefore we are modeling optimization criteria adopted by different actors of the system, using the same objective function, under the same constraints. This can be done as in OPT2 correctly, however when we add transfer and bus capacity constraints, these criteria have different constraints. This lead us to consider a bilevel mathematical programming formulation for the TNDP.

3.4.1 Bilevel mathematical programming

Bilevel mathematical programs [12, 25, 49] model scenarios with the following characteristics:

- The decisions are taken by two different agents, who constitute a hierarchy.
- Each agent may have different objectives and constraints and can exercise direct control over only certain variables.

- The agent corresponding to the upper level of the hierarchy should take decisions which: (a) constraint the decisions of the agent corresponding to the lower level and (b) need to anticipate the reaction of the lower level.

The decision making process modeled by a bilevel mathematical program can be seen as carried out in two sequential stages [49]: first, the higher level announces his plan of action and second, the lower level reacts rationally to that plan. The plan announced by the higher level is taken as exogenous data by the lower level, that independently optimizes his plan of action according to his goals and limitations, disregarding the goals of the higher level. Decisions of the lower level influence decisions of the higher level, since lower level variables may be present at constraints and objective function of the higher level.

The general formulation of a bilevel programming problem (BLPP) is the following [25]:

$$\min_{x \in X, y} \quad F(x, y) \tag{3.36}$$

$$\text{s.t.} \quad G(x, y) \leq 0, \tag{3.37}$$

$$y \in \arg\min_{y'} f(x, y') \tag{3.38}$$

$$\text{s.t.} \quad g(x, y') \leq 0, \tag{3.39}$$

where x, y and y' are real vectors, but they may be integer of binary as well.

In BLPP we identify upper level variables x and y, objective function F and constraint G, while their counterparts corresponding to the lower level are y', f and g respectively. The set X imposes an additional constraint to the upper level variables, independent of the lower level ones. On the other hand, to verify constraint G for a given x, we need to know the solution y of the lower level problem (which does not know constraint G, instead it has its own constraint g). Observe that a special constraint of the upper level states that feasible solutions should be optimal solutions to the optimization problem corresponding to the lower level; in fact, the original denomination for this kind of problem was "mathematical programs with optimization problems in the constraints" [25]. Special cases of BLPP have been used to model scenarios having the characteristics named at the beginning of this section, in areas related to economics, transportation and engineering [12, 25]. An extensive annotated bibliography on this topic can be found in [32].

Hereafter, for sake of simplification we use the following notation to refer to BLPP:

$$\min_{x \in X, y} \quad F(x, y) \tag{3.40}$$

$$\text{s.t.} \quad G(x, y) \leq 0, \tag{3.41}$$

$$\min_{y} f(x, y) \tag{3.42}$$

$$\text{s.t.} \quad g(x, y) \leq 0, \tag{3.43}$$

where we dropped the reference to argmin and the prime symbol from the lower level variables. Observe that a trivial relaxation of this problem is as follows:

$$\min_{x \in X, y} \quad F(x, y) \tag{3.44}$$

$$\text{s.t.} \quad G(x, y) \leq 0, \tag{3.45}$$

$$g(x, y) \leq 0, \tag{3.46}$$

which has the form of a conventional one-level mathematical programming formulation.

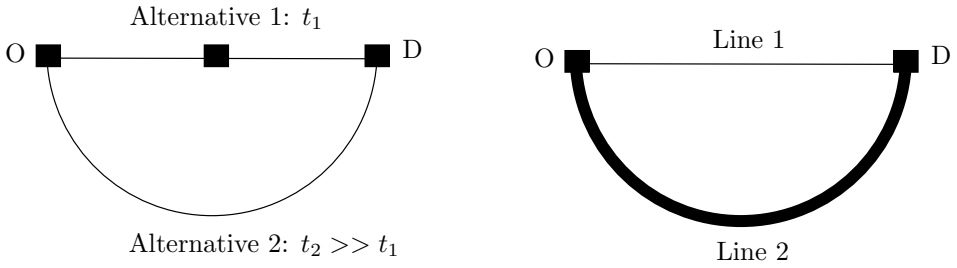
Concerning the computational complexity of BLPP, it has been proved that even in its simpler form (the case where both levels are linear) the problem belongs to the NP-hard class. Anyway, solution methods that guarantee to find the global optimum for that variant have been proposed [12, 25].

3.4.2 Bilevel formulation for the TNDP

If we add the transfer and bus capacity constraints directly to OPT2, the implicit assignment of flows will be done respecting them, modeling a situation that is not coherent with the reality.

Concerning constraint (3.33), in the optimal solution the demand may result assigned to trajectories that do not involve transfers, when there are faster trajectories that involve transfers. Observe that in reality, by enforcing transfer constraints, it is the planner who wants to avoid transfers, while the users ignore them in the assignment model of optimal strategies. Figure 3.5(a) illustrates this situation. Assume a single OD pair which has two alternatives to travel: the first one involves one transfer but has a small travel time t_1 while the second one does not involve transfers but has a very high travel time $t_2 >> t_1$; also assume $\Delta^1 = 0.5$, meaning that at least the half of the total demand should travel directly (without transfers). If we add constraint (3.33) directly to formulation OPT2, in the resulting optimal solution at least the half of the demand corresponding to that OD pair will be assigned to the alternative 2 which has a very long travel time, when a more realistic assignment will route the entire demand to alternative 1 (which includes one transfer).

Concerning capacities, if we add constraint (3.35) directly to formulation OPT2, we may obtain an optimal solution having an assignment of demand that is not realistic. As an example (Figure 3.5(b)), consider a set of passengers corresponding to the same OD pair, having two alternatives to travel: Line 1 has a very small on-board travel time and low frequency (therefore low capacity), while Line 2 has higher frequency (therefore higher capacity) but also has a very higher on-board travel time. In this case, the solution according to the optimal strategies assignment model with the bus capacity constraint added directly, will split the demand into two groups that consider a priori different sets of attractive lines: the first group will use the whole capacity of Line 1, while the second one will use Line 2 with a very higher overall travel time. Observe that a more realistic behavior for this second group of passengers would be to use Line 1 anyway, which entails to wait for the next bus with available capacity; this leads to the concept of assignment under hypothesis of congestion which is out of the scope of this thesis (see Section 2.3). We resume this issue in the next subsection corresponding to bus capacity constraint considerations.



(a) Demand covering (b) Bus capacity

Figure 3.5: Adding constraints directly to OPT2

An alternative to model in a realistic manner the inclusion of transfer and bus capacity constraints, is to consider the bilevel nature of public transportation systems. Observe that the variables involved in those constraints (in particular v) represent decisions of the users that are subject to decisions of the planner; therefore, he can enforce those constraints by deciding appropriate lines and frequencies (x and y respectively) so that the users will act according to the hypothesis of the optimal strategies assignment model. This means that they will perform transfers when they consider necessary to do so and they will perceive unlimited bus capacities. The bilevel mathematical programming formulation (3.47)-(3.61) called OPT3 captures this situation.

We can observe that in OPT3, the objective functions of both levels are the same, however, the decision variables are different; while the upper level objective function models the interest of the users (routes x with frequencies y, determined by the planner), the lower level objective function models their behavior (flows v and waiting times w, determined by the assignment model). Constraints (3.48)-(3.52) involve only variables of the upper level; they model constraints that the planner should take into account, independently of the reaction of the users to his decisions. Constraints (3.60)-(3.61) involve variables that belong to both levels or to the lower level only; they model constraints that the planner should take into account after knowing the reaction of the users (given by the lower level problem) to his decisions. The lower level problem represents the assignment model of optimal strategies, where constraints (3.56) and (3.57) state that only the routes and frequencies enabled by the planner can be used.

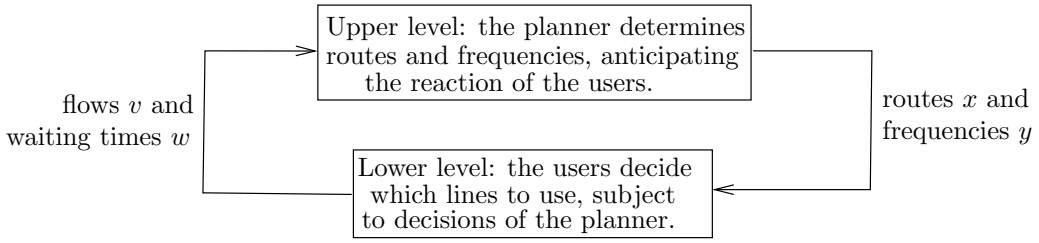


Figure 3.6: Bilevel structure of the TNDP

Figure 3.6 illustrates the interdependence between the decisions of the planner and the users, modeled by OPT3. Bilevel programming has been used to model many urban transportation problems, where relationships similar to the one encountered in the TNDP

need to be represented, usually involving a central planner and a set of users [92].

$$\min_{x,y,v,w} \quad \sum_{k\in K}(\sum_{a\in A} c_a v_{ak} + \sum_{n\in N} w_{nk}) \tag{3.47}$$

$$\text{s.t.} \quad \sum_{r\in R} 2 \sum_{f\in\Theta} \theta_f y_{rf} \sum_{e\in E_r} c_e \leq B, \tag{3.48}$$

$$\sum_{r\in R}\sum_{f\in\Theta} y_{rf}\theta_f \Lambda_{er} \leq \kappa_e \qquad \forall\ e \in E, \tag{3.49}$$

$$\sum_{f\in\Theta} y_{rf} = x_r \qquad \forall\ r \in R, \tag{3.50}$$

$$x_r \in \{0,1\} \qquad \forall\ r \in R, \tag{3.51}$$

$$y_{rf} \in \{0,1\} \qquad \forall\ r \in R, f \in \Theta, \tag{3.52}$$

$$\min_{v,w} \sum_{k\in K}(\sum_{a\in A} c_a v_{ak} + \sum_{n\in N} w_{nk}) \tag{3.53}$$

$$\text{s.t.} \sum_{a\in A_n^+} v_{ak} - \sum_{a\in A_n^-} v_{ak} = b_{nk} \qquad \forall\ n \in N, k \in K, \tag{3.54}$$

$$v_{ak} \leq \theta_{f(a)} w_{nk} \qquad \forall\ a \in A_n^{W+}, n \in N, k \in K, \tag{3.55}$$

$$v_{ak} \leq \delta_k x_{r(a)} \qquad \forall\ a \in A, k \in K, \tag{3.56}$$

$$v_{ak} \leq \delta_k y_{r(a)f(a)} \qquad \forall\ a \in A^W, k \in K, \tag{3.57}$$

$$v_{ak} \geq 0 \qquad \forall\ a \in A, k \in K, \tag{3.58}$$

$$w_{nk} \geq 0 \qquad \forall\ n \in N, k \in K, \tag{3.59}$$

$$\sum_{k\in K} \sum_{a\in\cup_{i=1..s} A_i^D} v_{ak} / \sum_{k\in K} \delta_k \geq \Delta^s \qquad \forall\ s \in [1..\tau], \tag{3.60}$$

$$\sum_{k\in K} v_{ak} \leq \sum_{f\in\Theta} y_{r(a)f}\theta_f \omega \qquad \forall\ a \in A^V. \tag{3.61}$$

Observe that additional constraints (other than those concerning transfers, infrastructure and bus capacity) impact in different ways on OPT3. For example, we may want to add a constraint on the maximum number of lines, expressed as

$$\sum_{r\in R} x_r \leq L, \tag{3.62}$$

where L is a positive integer value that states the maximum number of lines in the solution. This constraint represents decisions of the planner that do not need to know the corresponding reaction of the users. Moreover, the planner may want to design a system that guarantees a maximum waiting time for any user at any stop; this leads us to the following constraint

$$\sum_{a\in A_n^{W+}} v_{ak} \geq w_{nk}/\epsilon \quad \forall\ n \in N, k \in K, \tag{3.63}$$

where ϵ is an imposed upper limit on the waiting time, expressed in time units. Observe that both constraints (3.62) and (3.63) belong to the upper level of OPT3. However, the

first one does not require a bilevel model in order to be included, since it involves only variables corresponding to decisions of the planner; by contrast, the second one involves variables controled by the users, so its modeling is analog to that one required when including the bus capacity constraint.

Bus capacity constraint considerations

The lower level of formulation OPT3 assumes that every user can board any bus of any line selected as part of his optimal strategy according to the assignment model [110]; in other words, it is assumed that the buses have unlimited capacity. If we consider the capacity of the bus in the hypothesis of the assignment model, we should take into account that the users may behave in a way different from as they do in the uncapacitated case. This approach leads to an assignment that models congestion, whose formulation and solution are considerably more complex than the case without congestion, since an equilibrium problem should be considered [19, 30, 55].

In OPT3, we consider the capacity constraint of the buses with a different approach: the optimization model should ensure sufficient capacity on the lines that the users desire to use. This implies that the bus capacity is not taken into account in the assignment sub-model; instead, it is included as a constraint of the route optimization model. This approach has been used in previous works concerning the TNDP and related problems [9, 26, 73]. It is worth mentioning that there are cases where it is not possible to ensure the desired capacity on the lines, given the high demand that may have certain lines and the technical limitations of the transportation mode, which imposes a maximum allowed frequency. In these cases it is necessary to include the bus capacity constraint into the assignment model, as it is done in [47] in the context of a variant of the TNDP and [52] that proposes a frequency optimization model. To the best of our knowledge, there are no published studies concerning a comparative evaluation of the performance of the routes and frequencies of a public transportation system, which has been designed according to the two alternatives mentioned above for including the bus capacity constraint.

3.4.3 Alternatives to solve the bilevel formulation for the TNDP

According to [118], the problem stated by OPT3 can be classified as a Discrete Continuous Linear Bilevel (DCLB) problem, i.e., linear bilevel with discrete variables in the upper level and continuous variables in the lower level. In this section we mention alternatives to solve OPT3 exactly, keeping in mind that in principle there is not an efficient standard solution method for this kind of problem, even if the set of possible routes is small.

The algorithm of Moore and Bard described in [12] could be used to solve OPT3 exactly. That algorithm guarantees to find the global optimum if all the variables controlled by the higher level are discrete and all the variables controlled by the lower level are continuous. This is the case of OPT3. The algorithm performs a branching on the (discrete) higher level variables, uses as subroutine an algorithm to solve a linear bilevel problem only with continuous variables at both levels and applies specific fathoming rules. Computational results involving instances comprising up to 40 variables are reported in [12]. With the increase of computer speed it is likely that today larger instances can be solvable.

A different alternative to solve OPT3 is based on the fact that the lower level of that formulation is linear with continuous variables only. Thus, the lower level problem can be substituted by the optimality conditions given by its constraints, the constraints of its dual and the complementary slackness conditions. The dual [13] of problem (3.53)-(3.59) can be expressed as

$$\max_{\pi,\lambda,\mu,\nu} \sum_{k\in K}\sum_{n\in N} b_{nk}\pi_{nk} \; - \; \sum_{k\in K}\sum_{a\in A} \delta_k x_{r(a)}\lambda_{ak} \; - \; \sum_{k\in K}\sum_{a\in A^W} \delta_k y_{r(a)f(a)}\mu_{ak} \tag{3.64}$$

s.t.

$$\pi_{ik} - \pi_{jk} - \lambda_{ak} \leq c_a \qquad \forall\, a=(i,j) \in A - A^W, k \in K, \tag{3.65}$$

$$\pi_{ik} - \pi_{jk} - \lambda_{ak} - \mu_{ak} - \nu_{ak} \leq c_a \qquad \forall\, a=(i,j) \in A^W, k \in K, \tag{3.66}$$

$$\sum_{a\in A_n^{W+}} \theta_{f(a)}\nu_{ak} \leq 1 \qquad \forall\, n \in N, k \in K, \tag{3.67}$$

$$\lambda_{ak} \geq 0 \qquad \forall\, a \in A, k \in K, \tag{3.68}$$

$$\mu_{ak}, \nu_{ak} \geq 0 \qquad \forall\, a \in A^W, k \in K. \tag{3.69}$$

Let s^1_{nk}, s^2_{ak} and s^3_{ak} be slack variables associated to inequality constraints (3.55), (3.56) and (3.57) respectively. Analogously, let t^1_{ak}, t^2_{ak} and t^3_{nk} be slack variables associated to inequality constraints (3.65), (3.66) and (3.67) respectively. Then, the complementary slackness conditions are

$$s^1_{nk}\nu_{nk} = 0 \qquad \forall\, n \in N, k \in K, \tag{3.70}$$

$$s^2_{ak}\lambda_{ak} = 0 \qquad \forall\, a \in A, k \in K, \tag{3.71}$$

$$s^3_{ak}\mu_{ak} = 0 \qquad \forall\, a \in A^W, k \in K, \tag{3.72}$$

$$t^1_{ak}v_{ak} = 0 \qquad \forall\, a \in A - A^W, k \in K, \tag{3.73}$$

$$t^2_{ak}v_{ak} = 0 \qquad \forall\, a \in A^W, k \in K, \tag{3.74}$$

$$t^3_{nk}w_{nk} = 0 \qquad \forall\, n \in N, k \in K. \tag{3.75}$$

Expressions (3.70)-(3.75) can be linearized by applying the method proposed in [49] which uses the disjunctive nature of the complementary slackness conditions and proposes to substitute each product xy by equations

$$x \leq Mz,$$
$$y \leq (1-z)M,$$

where z is a binary variable and M is a sufficiently high positive value. Thus we can substitute in OPT3 problem (3.53)-(3.59) by its constraints (3.54)-(3.59), the constraints of its dual (3.65)-(3.69) and the linearized version of the complementary slackness conditions (3.70)-(3.75). In this way we can transform the original DCLB formulation into a MILP one. Note that a large number of new binary variables are introduced in order to obtain this one-level formulation. According to [12] in principle this method is not necessarily efficient, since some limited experiments suggest that a large portion of the search tree

has to be enumerated. However, further studies that apply a similar approach to network design problems ([67] and [84] which is co-authored by the author of this thesis) have shown that cases of moderate sizes can be solved.

3.5 Numerical experiments

In this section we perform numerical experiments using the formulations developed. Our goal is to investigate the possibilities of using the models, by applying directly a general purpose MILP solver. Note that we are not proposing any concrete optimization approach to the problem based on the formulations developed. More precisely, the aim of the experiments is to produce results to answer the following questions:

1. Up to which size (in terms of number of vertices, edges and commodities) are instances solvable to optimality?
2. Can the formulations be used in a process of decision making in the context of a real instance of the problem?

In order to answer 1, we use small-sized instances either generated specifically for this study as well as taken from the literature. For 2, we use the test case relative to the city of Rivera (Appendix A). The experiments were ran using CPLEX 10 under Linux, in a Dual Core computer of 2.8 GHz with 2 GB of RAM memory, that we call machine 1. The experiments that involved large-sized models were ran using CPLEX 12, in a Core i7 computer of 3.4 GHz with 16 GB of RAM memory, that we call machine 2.

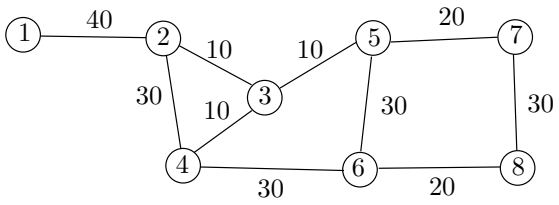
3.5.1 Small instances

In this experiment we try to solve exactly the problem stated by formulation OPT2 without including transfer arcs; therefore the set R should contain all the possible routes defined according to the hypothesis stated in Section 2.1.3. We use two test instances (Figure 3.7):

- Small: An instance specially generated for this study, comprising 8 vertices, 10 edges and 4 OD pairs. Figure 3.7(a) shows its corresponding infrastructure graph G and OD pairs K; edge on-board travel time is expressed in minutes while demand quantity is expressed in trips per minute.
- Wan and Lo: Taken from [120], comprising 10 vertices, 19 edges and 9 OD pairs. Edge on-board travel time is expressed in minutes while demand quantity is expressed in trips per hour in Figure 3.7(b).

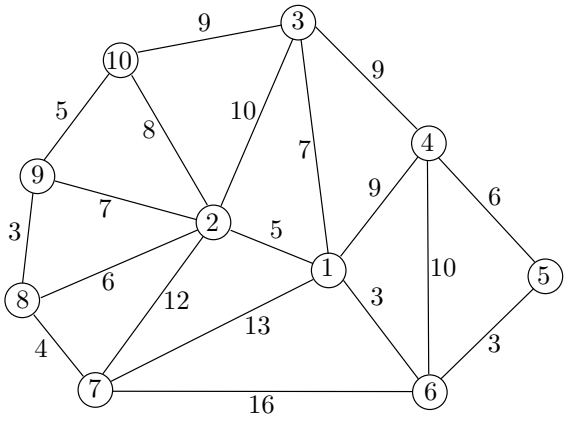
To solve Small, the size of the set R of all possible routes is 79 after eliminating all routes having at least one endpoint that is neither origin nor destination of any OD pair; note that by eliminating these routes we are not excluding the optimal solution, since we are not considering transfers. We use three different values for the fleet size B (8, 20 and 60 buses) and the set of possible frequencies was set as $\Theta = \{1/60, 1/30, 1/5\}$, expressed in buses per minute. The numbers of variables and constraints of the corresponding model are 8,212 and 13,880 respectively. For Wan and Lo, we imposed an upper limit on route duration, since the number of all possible routes is very high (3580). By doing this we

obtain a set R comprising 608 routes, while we use a set of frequencies $\Theta = \{3, 9, 15, 20\}$, expressed in buses per hour. For setting B we use the value of the optimal solution obtained in [120], equal to 995. The resulting model has 199,465 variables and 393,090 constraints.



O	D	Demand
3	1	1.0
3	8	2.0
4	7	0.5
6	1	0.5

(a) Small



O	D	Demand
2	10	200.0
3	2	150.0
4	7	800.0
5	8	350.0
6	9	600.0
7	6	250.0
8	3	400.0
9	4	450.0
10	5	500.0

(b) Wan and Lo

Figure 3.7: Small-sized test cases

The results are presented in Table 3.2, where *Opt* denotes the optimal value (or best found), LR denotes the optimal value of the linear relaxation and *Gap* is the percentage gap between these two values, relative to the last one. R^* is the optimal set of routes, $1/f$ refers to the average of headways (inverse of frequency) of each solution (note that average headways do not necessarily coincide with the inverse of any value in Θ), while T is the execution time expressed in seconds (in machine 1). The rows tagged with † indicates that a time limit of 4 hours was imposed. We can observe that only the case Small with low values on the fleet size constraint was solvable to optimality. For a high value on that constraint, the optimal value was not attained within the imposed time limit. We may expect this behavior since the fleet size constraint bounds the size of the feasible space. Also note that the gap decreases according to the increase in the fleet size. Another expectable result is the increase of number of routes and decrease of average headway, according to the increase in the available fleet size. For the instance of Wan and Lo, it was necessary to reduce the size of the route set R, since the memory was not sufficient to load the model generated using the whole set of feasible routes. This behavior is also expected, since the cardinality of this set grows in a super-polynomial order as a function of the size of graph G.

Instance	*Opt*	*LR*	*Gap*	$\|R^*\|$	$1/f$	T
Small $B = 8$	385.00	253.03	52	3	60	1
Small $B = 20$	291.50	239.60	21	5	42	547
Small $B = 60$ †	254.20	236.69	7	7	31	-
Wan and Lo †	1778.19	1170.44	52	6	1/3	-

Table 3.2: Results of OPT2 applied to instances Small and Wan and Lo

3.5.2 Real test case

In this experiment, we applied OPT2 to the case of the city of Rivera (Appendix A); transfer arcs were not included in order to be consistent with the behavior of the users of such system. We did not intend to solve this case to optimality, instead we wanted to explore the applicability of the model to a given situation of decision making in the context of a real case. More specifically, the model was used to select the optimal subset of routes from a given pool of routes R and to determine their optimal frequencies. That pool might for example include the routes of the current system and other alternative routes identified by the planner; the model may be used in this case to decide whether it is convenient to replace an existing route by a new one.

In order to generate a pool of routes, we ran the Pair Insertion Algorithm (PIA, Chapter 4) three times, for different values of its maximum route duration parameter. We selected routes randomly from those executions, obtaining a pool R comprising 48 routes. We used the current solution of Rivera as reference to configure the set of possible frequencies Θ and the value B of the fleet size constraint; the first one includes the values of frequencies used by the lines of Rivera ($\{1/60, 1/40, 1/30, 1/20\}$, expressed in buses per minute), while the second one is equal to 25.65 (see Section 5.5.4). The model generated with those data to run OPT2 has 11,990,000 variables (from which 192 are binary) and 9,528,450 constraints.

We imposed a time limit of 4 hours on machine 2, obtaining an integer feasible solution with a 12% relative MIP gap and cost equal to 587.00. The solution comprises 16 routes taken from the three runs of PIA executed to construct the provided pool R. The model selected 16 routes with low values of frequencies ($1/f$=51), instead of selecting less routes with higher frequencies. This is considered by the model as the best way of exploiting the available fleet size; observe that high frequency lines require a higher fleet size. Moreover, low frequencies do not necessary imply low waiting time, because this value is computed in terms of the frequencies of potentially many lines (expression (2.1)). We performed an additional experiment by adding the constraint (3.62) of maximum number of lines to OPT2, with $L = 15$. In this case, the optimal solution was found within the time limit, with cost equal to 590.68. The solution comprises 15 routes with higher frequencies ($1/f$=45). Thus, we can observe that the model increased the frequencies in order to improve the overall travel time, when the total number of routes is bounded.

In summary, we can observe that the model was able to solve the particular application of the TNDP posed on this experiment. We showed that OPT2 could be used in a scenario of decision making concerning a small-sized real case. We note that the routes of the pool provided by us are not intended to be directly practicable in the real city; the goal of this

experiment is only to show that the resulting model can be solvable. Some improvements to the implementation of the model could be performed, in order to make more efficient the application of a MILP solver. For example, we can implement a preprocessing technique that substitutes segments of routes by a single arc, thus reducing the total number of variables of the model. We observed that adding more routes to the pool R reduced considerably the chances of solving the problem, mainly due to high memory requirements.

3.6 Conclusions and future work

We have proposed mathematical programming formulations to model the TNDP. The formulations are based on definitions of two graph models, an infrastructure and a trajectory graph, that are used to represent decisions of the planner and the users, respectively. The need for including both graphs into the same model is not usually recognized in the literature relative to the TNDP. We start from a base formulation, which includes an existing assignment sub-model that considers multiple routes and the waiting time in the behavior of the users. This last aspect of the assignment problem has not been considered in the existing explicit formulations for the TNDP. Then we add constraints that represent important aspects of the problem, like transfer, infrastructure and bus capacity constraints. We study the impact of adding these constraints in the mathematical structure of the formulation, based on concepts of bilevel mathematical programming; alternative solution methods for the bilevel formulation are mentioned.

By means of numerical experiments we give an idea of the size of problem instances that are solvable by using a standard MILP solver, in terms of number of vertices and edges of the infrastructure graph and number of OD pairs. We also show that the value of the fleet size constraint plays an important role when solving the problem. Moreover, we show that the base formulation could be used in scenarios of decision making with a real test case, relative to a small city; although the model generated for this particular application is very big (several millions of variables and constraints), it is solvable with a reasonable distance to optimality.

We do not propose an specific algorithm to solve the formulations, therefore it is an interesting future task to continue the research on this direction; for instance by applying a column generation approach as it is done in [15, 107]. Another line of future research concerning this part of the thesis, is the investigation of the applicability of the solution alternatives mentioned in Section 3.4.3 to the bilevel formulation including transfer and bus capacity constraints; in particular, the special structure of that formulation (the objective functions of both levels are the same) may be exploited in order to improve the efficiency of the solution methods.

Chapter 4

Route construction algorithm[†]

In this chapter we present a constructive algorithm to generate a set of routes to solve the TNDP. The algorithm is specially designed to produce a set of routes that fulfils demand covering constraints (Section 2.3), while taking into account the interests of both users and operators. The motivations for developing this algorithm are:

- The need of a heuristic to obtain approximate solutions, given the high combinatorial complexity of the problem [15, 65, 107].
- Usually a metaheuristic needs an initial solution to start the search and requires to evaluate the objective function many times. In the case of the TNDP, that evaluation entails an invocation to the assignment model (Section 2.2), which is computationally intensive. For this reason it is desirable to have an initial feasible solution as good as possible, without requiring repeated invocations to the assignment model.
- It is desirable to have an algorithm whose logic is simple and understandable to be used in an interactive way by the planner.

The general structure of the proposed algorithm is inspired in the Route Generation Algorithm (RGA) [10], where its original expansion of routes by inserting individual vertices is replaced by a strategy of insertion of pairs of vertices. The proposed algorithm, called Pair Insertion Algorithm (PIA) can be used to generate initial solutions for a local improvement or evolutionary algorithm, as well as to complete an unfeasible solution with respect to demand covering constraints. Numerical results comparing PIA with RGA over a real test case show that both algorithms produce solutions with similar quality from the users viewpoint (in terms of on-board travel time), while PIA produces better solutions from the operators viewpoint (in terms of number of routes and total route duration) and requires a higher execution time. Since the TNDP arises in a context of strategic planning, a solution that reduces the operation cost of the system is highly desirable, even though it takes more time to be computed. The experimental study of the proposed algorithm also shows its ability to produce diverse solutions in both decision and objective spaces; this is a useful property when looking at the use of PIA as a subroutine in the context of another algorithm such as metaheuristics, in particular for a multi-objective problem like TNDP.

[†] Most of the content of this chapter was published in [85].

We use PIA jointly with a frequency optimization model, to obtain a complete solution (routes and frequencies) for the TNDP, and we compare these results with optimal values.

4.1 Introduction

A key component in the overall planning of a public transportation system is the network design, where a set of routes is defined over the street network. According to [18], network design is the first of the five stages of a systematic decision sequence, followed by frequency setting, timetable development, bus and driver scheduling. Decisions at network design level are usually taken for a long term horizon, in the context of strategic planning [33]. The transit network directly determines characteristics of the public transportation system with respect to the users' interest such as geographical accessibility and travel time; alongside with the frequencies, it also defines an important component of the cost for the operators. Once the transit network is defined, all the subsequent decisions about timetable development, and bus and driver scheduling are conditioned by it, so the overall cost of the public transportation system highly depends on the transit network [18].

The Transit Network Design Problem (TNDP) aims to find a set of routes with their corresponding frequencies, optimizing the objectives of users and operators [9]. Main problem data are the street network and the demand of trips between different points of the city. Constraints refer usually to demand covering, required level of service and resource availability. Frequencies are included in the TNDP as decision variables because they also have a direct influence in the cost structure of both users (determining the waiting time) and operators (defining the required fleet size).

The TNDP is a hard to solve combinatorial problem [65], given the discrete nature of some of its variables (those that represent routes). It is also difficult to formulate with a mathematical programming approach [20]. For all these reasons, most existing approaches to solve it rely on approximative methods, i.e., heuristics and metaheuristics [10, 44, 109, 113]. Most of these methods use an specific purpose algorithm to explicitly construct a set of routes, which is not always feasible with respect to demand covering constraints (Section 2.3).

A few works exist in the literature about heuristic algorithms to construct a set of routes for the TNDP, while ensuring demand covering feasibility [10, 65, 94]. Only the Route Generation Algorithm (RGA) [10] generates, starting from an empty solution, a set of routes that covers the whole demand, while considering some aspects of interest from both users and operators points of view. RGA was designed taking into account several desirable properties and design principles; in this sense, it can be considered as a heuristic that incorporates a deep knowledge of the problem from the application viewpoint. It includes a mechanism to explicitly cover the demand (with or without transfers). However, the cost for the operators (represented by number of routes and total route mileage) of the resulting set of routes produced by RGA remains high when the requirement of demand covering is increased [10].

In this work we take some ideas from RGA and we add new ones to propose an algorithm that produces solutions that are more convenient for the operators in comparison with RGA, while maintaining almost the same cost for the users (in terms of travel time). Those solutions are highly desirable, since they reduce the operation cost of the transit

system, allowing to operate a more sustainable system while not degrading its quality from the users viewpoint. The algorithm proposed addresses the problem of the fulfilment of demand covering constraints, while taking care in producing solutions that are convenient for both users (low on-board travel time) and operators (low number of routes and total route mileage). A solution constructed by the proposed algorithm can be used as a starting point for a local improvement method or as an initial solution for an evolutionary algorithm, in order to improve its quality. The algorithm can be also used to complete an unfeasible solution with respect to demand covering constraints. We have performed computational experiments which are based on a real test case that constitutes a reference to compare the results produced by the algorithm.

The chapter is organized as follows. In Section 4.2 we give some definitions and notation used in the following sections. A brief literature review specially focused on approximative methods and route set construction algorithms for the TNDP is given in Section 4.3, as well as the motivation of this work. A detailed description of the proposed route set construction algorithm is presented in Section 4.4. Numerical results of an experimental study and their analysis are shown in Section 4.5 while conclusions and future work are given in Section 4.6.

4.2 Definitions and notation

We assume that an infrastructure graph $G = (V, E)$ and an origin-destination matrix D are given; we refer to Chapter 2 for the definition of these elements. For simplicity, as we did it in Section 3.2, we adopt the simplified model for representing the routes (Section 2.1.3) and we assume that all vertices are of type street, stop and centroid at the same time.

We only work with the routes $R = \{r_1, \ldots, r_r\}$ of a solution for the TNDP (Section 2.3); frequencies are not taken into account. The interests of users and operators are represented by functions Y_1 and Y_2 respectively. The former is defined as

$$Y_1(R) = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} d_{ij} t_{ij}(R)/t^*_{ij} \,, \tag{4.1}$$

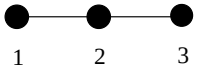
where $t_{ij}(R)/t^*_{ij}$ represents (for passengers traveling from i to j) a deviation ratio of the minimum on-board travel time using routes of R, from the minimum possible value (independent of any set of routes). According to this, $t_{ij}(R)$ is calculated by using an all-or-nothing assignment approach (Section 2.2), and t^*_{ij} is calculated as the cost of the shortest path in G between i and j. We assume that passengers apply a transfer avoidance criterion (as it is done in [8, 60]); therefore, when assigning the demand, a trajectory with higher on-board travel time but lower number of transfers is preferred. In the remaining part of this chapter, when we refer to shortest paths in G and cost of a route r, it is always with respect to the values of on-board travel time represented by c_e for every edge $e \in E$; in this way $cost(r) = \sum_{e \in r} c_e$.

Operators' interests are represented by

$$Y_2(R) = \sum_{r_k \in R} t_k \,, \tag{4.2}$$

R	$D_0(R)$	$D_0(R) \geq D_0^{min}$	$D_{01}(R)$	$D_{01}(R) \geq D_{01}^{min}$
{(1,2)}	0.10	×	0.10	×
{(1,2);(2,3)}	0.20	×	1.00	✓
{(1,2,3)}	1.00	✓	1.00	✓

Table 4.1: Demand covering for three different sets of routes



(a) Graph

0	5	40
5	0	5
40	5	0

(b) Matrix

Figure 4.1: Illustrative example

which is the total duration of routes in R (equivalent to the total route mileage), where $t_k = 2\sum_{e \in r_k} c_e$ is the duration (round-trip time) of route r_k. Low values in both functions (4.1) and (4.2) are considered in this work as a desirable property for a set of routes R which is intended to be part of a good solution (routes and frequencies) for the TNDP.

We adopt the following notation to express the demand covering constraints. For a given set of routes R, $D_0(R) \in [0, 1]$ is the proportion of the total demand $D_{tot} = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} d_{ij}$ covered by routes in R directly (without transfers). Similarly, $D_{01}(R)$ is the proportion of D_{tot} covered by routes in R directly or indirectly (one transfer, at most). D_0^{min} and D_{01}^{min} are constant values, which constrain $D_0(R)$ and $D_{01}(R)$ respectively as

$$D_0(R) \geq D_0^{min} \,, \tag{4.3}$$

$$D_{01}(R) \geq D_{01}^{min} \,. \tag{4.4}$$

Table 4.1 shows for the illustrative case of Figure 4.1 and for three different sets of routes R, the corresponding values of $D_0(R)$ and $D_{01}(R)$ and results of checking the fulfilment of demand covering constraints when $D_0^{min} = 0.75$ and $D_{01}^{min} = 1.00$.

4.3 TNDP and route construction

The TNDP is a hard to solve optimization problem. Its exact resolution has several difficulties [9, 20], namely, high combinatorial complexity, a multi-objective nature [65] and the requirement of an assignment sub-model [33].

It has been treated almost exclusively with approximative methods. Its combinatorial complexity prohibits the exhaustive enumeration of all feasible solutions. Mathematical programming formulations face the difficulty of modeling the assignment component; existing work using this approach has made simplifications to it [15, 107, 120]. We refer to Sections 1.2 and 3.1 for a more detailed discussion about these characteristics of the problem.

Existing approximative methods for the TNDP can be classified in two groups:

1. Heuristics. Classical approximative methods, either constructive or local improvement procedures, as well as combinations of them [9, 10, 109]. Another kind of heuristics for the TNDP consists of selecting the best possible set of routes from a previously generated pool of candidates [65].

2. Metaheuristics. Modern approximative methods that implement efficient mechanisms to explore the search space. The application of metaheuristics to the TNDP has been concentrated in using Genetic Algorithms [20, 94, 101, 103, 113], but some works also explore the use of Tabu Search [44] and Simulated Annealing [43].

When solving the TNDP with some approximative methods, routes have to be explicitly designed by using a route construction algorithm. Several routes must be generated, which are then grouped to form a set of routes that fulfils the constraints of the problem. An important type of these constraints are the *demand covering constraints*.

A few works exist in the literature about heuristic algorithms to construct a set of routes for the TNDP, while ensuring demand covering feasibility. In [10], a greedy constructive algorithm that generates a set of routes from scratch is proposed; [65] uses a non-linear set covering formulation to select a subset of routes from a previously generated pool of candidate routes. In [44], an optimization model that includes the uncovered demand in the objective function is used, while the algorithms proposed in [94, 101] do not guarantee demand covering at the end of the execution.

The Route Generation Algorithm (RGA) [10] is the only constructive algorithm that generates from scratch a set of routes, ensuring the fulfilment of demand covering constraints; it also takes care of the interests of users and operators in the produced results. RGA also allows to specify a predetermined set of routes as an initial partial solution. It proceeds by iteratively adding routes to the solution under construction. Routes are generated by using the shortest path in G between vertices with high demand. Additional vertices are then inserted in these routes (expansion of routes) according to a pre-specified criterion. The algorithm ends when the set of routes under construction fulfils demand covering constraints (4.3) and (4.4). Figure 4.2 shows a seudo-code of RGA.

A key step in RGA is the expansion of routes (procedure **ExpandRoute**), where the algorithm takes advantage of a previously existing route to cover the demand between vertices which are close to it and vertices which are already included into the route. Thus, the expansion of a route considers the insertion of vertices on it, taken from a set of candidate vertices. A feasible candidate is a vertex v which is at distance 1 (measured in number of edges) from the route r and which fulfils the following constraints (as explained in [10]):

1. It does not already belong to r.

2. It still has a high percentage of its total originating demand left uncovered after previous insertions in other routes.

3. The resulting route (after insertion of v in r) does not become circuitous. This is determined by comparing the on-board travel time between the extreme vertices of the route, against the travel time between those vertices over the graph G (independently of any route).

```
procedure RGA(in D_0^min, D_01^min, out R);
  R ← ∅; D_0(R) ← 0; D_01(R) ← 0;
  l ← List of pairs of vertices (i, j) of G with d_ij ≠ 0;
  while D_0(R) < D_0^min or D_01(R) < D_01^min do
     (u, v) ← Select (i, j) with maximum d_ij in l;
     r ← Create a route with the shortest path between u and v in G;
     ExpandRoute(r);
     R ← R ∪ {r};
     Delete from l pairs of vertices whose demand is covered directly by r;
     Update D_0(R) and D_01(R);
  end while;
  Filter routes in R;
  return R;
end RGA;
```

Figure 4.2: RGA, general structure

4. The ratio of the contributed incremental demand covered to the insertion cost (on-board travel time) exceeds a minimum value.

5. The required frequency of service on the resulting route does not exceed a maximum operationally implementable value. A preliminary assignment is performed (as described in the explanation of expression (4.1)) and the resulting flow is compared against a maximum allowable line capacity (determined by given parameters of maximum frequency and bus capacity).

6. The round-trip time of the resulting route does not exceed a maximum allowable value.

A route is expanded until the set of feasible candidates to be inserted is empty. Since after expansion, a route may include another existing route, a filter procedure is included at the end of the algorithm. Other elements of RGA (the initial number of routes, the order of expansion of routes and the use of k-shortest paths) were left aside on this simplified description; the essence of the algorithm is given by the strategy of the expansion of routes.

The computational experiments performed with RGA in [10] show that when demand covering requirements are increased, values of number of routes and total route mileage are significantly increased. For example, a change in D_{01}^{min} from 0.90 to 1.00 causes an increase of about 100% in the number of routes and 60% in the total route mileage. Despite the fact that the algorithm can produce good solutions from the users viewpoint, if these solutions have a high cost for the operators, they could result in a system that either is very expensive for the users (in terms of fares) or it may need a great amount of subsidies to cover the operation costs.

This issue motivated the research concerning this part of the thesis, where a new strategy of insertion of vertices on existing routes is proposed, inspired in the general structure of the Route Generation Algorithm.

4.4 Pair Insertion Algorithm

The Pair Insertion Algorithm is based in the observation that the expansion of routes is a key component in the overall design of RGA, and therefore it determines the quality of the solutions produced. Since only the insertion of individual vertices on existing routes is considered by RGA, the inter-zonal nature of the demand is not taken into account. This inter-zonal nature, which is given by the origin-destination matrix, is addressed on this work by considering the insertion of *pairs of vertices* on existing routes, therefore, covering directly the demand associated to them. In this way, vertices which are at distances higher than 1 from the route will be potentially inserted on it.

The basic principle of PIA is to connect pairs of vertices with high values of demand. The connection is made either by creating a new route based in the shortest path in G between those vertices, or by inserting both vertices in an existing route. Figure 4.3 outlines the main structure of the algorithm. It starts with an empty set of routes R, and iteratively seeks to cover the demand given by origin-destination matrix D. A list l of pairs of vertices whose demand is still not covered directly is maintained. At each iteration step, the pair of vertices (u, v) with the highest demand d_{uv} in l is selected and that demand is covered according to one of the two following possibilities:

1. Creating a new route, using the shortest path between u and v in G.

2. Inserting vertices u and v in suitable positions of a convenient route of R. It evaluates the cost of insertion of both u and v between all pairs of consecutive vertices in routes of R. The most convenient route and the most suitable positions for insertion of vertices u and v on it are those which minimize the cost increase in the solution under construction.

The lowest cost increase due to insertion of vertices u and v in a route of R according to case 2 is compared with the cost of the shortest path between u and v according to case 1; the best (less costly) case is selected and the algorithm proceeds. It ends when constraints of demand covering imposed by parameters D_0^{min} and D_{01}^{min} are fulfilled. Observe that the structure of the main loop of PIA is the same as the RGA's one. The difference is that where RGA always create a new route to cover the demand associated to the first element of l, PIA evaluates if that demand can be covered by expanding an existing route, thus trying not to add an additional route.

Figure 4.4 explains the computation of the most convenient candidate route built by insertion of a pair of vertices; we discriminate the cases when no vertex belongs to the route and when one vertex belongs to the route. A route r with $|r|$ vertices has $|r|+1$ possible positions for insertion of a single vertex; 1 denotes the position before the first vertex of the route and $|r|+1$ denotes the position after the last vertex. When insertion of vertex v is performed between two consecutive vertices v_i and v_{i+1} in r, it is connected to them by using the shortest paths in G between v_i and v, and between v and v_{i+1} respectively. If the resulting route after the insertion contains a loop, it is discarded as candidate.

In procedure **Candidate**, when trying to insert vertices into routes, two constraints are imposed: maximum duration (round-trip time) and maximum circuity factor, represented by parameters t_{max} and ρ_{max} respectively. The circuity factor ρ of a route r with extreme

```
procedure PIA(in D_0^min, D_01^min, in ρ_max, t_max, out R);
  R ← ∅; D_0(R) ← 0; D_01(R) ← 0;
  l ← List of pairs of vertices (i, j) of G with d_ij ≠ 0;
  while D_0(R) < D_0^min or D_01(R) < D_01^min do
     (u, v) ← Select (i, j) with maximum d_ij in l;
     r ← Create a route with the shortest path between u and v in G;
     r' ← Create a route by inserting u and v in the most suitable
           positions in the most convenient route r'' in R, by calling
           Candidate(u, v, R, ρ_max, t_max, r');
     if cost(r) < cost(r') − cost(r'') then
        R ← R ∪ {r};
        Delete from l pairs of vertices whose demand is covered directly by r;
     else
        R ← R ∪ {r'} − {r''};
        Delete from l pairs of vertices whose demand is covered directly by r';
     end if;
     Update D_0(R) and D_01(R);
  end while;
  Filter routes in R;
  return R;
end PIA;
```

Figure 4.3: PIA, general structure

vertices u and v, is defined in [10] as the ratio between the on-board travel time between u and v using r, and the cost of the shortest path between u and v in G (independent of any route), i.e., $\rho(r) = cost(r)/t^*_{uv}$. These constraints are imposed to limit the growth of the route, when several vertices have been inserted on it. Other constraints such as route capacity constraints can be easily incorporated to the model and implemented in the proposed algorithm.

Since the insertion of pairs of vertices on a route r may imply the insertion of a whole path P in r, it may be possible that there is an already existing route $r' \in R$ included in P. For this reason, there can be at the end of the main loop of PIA routes that are completely included in other ones. As in RGA, PIA has a filter procedure that eliminates these included routes, because we are interested in minimizing the number of routes and total route duration.

4.4.1 Rationale of the algorithm

The design of PIA is based in the following line of thought.

Probably the main objective in transit network design is to cover the demand in the best possible way, given a restriction in the available resources (however other objectives can be stated when designing a transit network [116]). Since the demand has an interzonal nature, expressed in the form of an origin-destination matrix, it has to be covered for *pairs of vertices*. For this reason, we consider the insertion of pairs of vertices on routes as a key idea of our algorithm.

The ideal solution from the users viewpoint is one that covers every non null element of

```
procedure Candidate(in u, v, in R, in ρ_max, t_max, out r');
  r' ← ∅; cost(r') ← ∞;
  for each r ∈ R do
    if u ∈ r then
      for each p ∈ [1..|r| + 1] do
        rAux ← Insert v in position p in r;
        if cost(rAux) < cost(r') and rAux respects ρ_max, t_max then
          r' ← rAux; Label r as r'';
        end if;
      end for;
    else if v ∈ r then
      for each p ∈ [1..|r| + 1] do
        rAux ← Insert u in position p in r;
        if cost(rAux) < cost(r') and rAux respects ρ_max, t_max then
          r' ← rAux; Label r as r'';
        end if;
      end for;
    else
      for each p1, p2 ∈ [1..|r| + 1] do
        r' ← Insert u and v in positions p1 and p2 respectively in r;
        if cost(rAux) < cost(r') and rAux respects ρ_max, t_max then
          r' ← rAux; Label r as r'';
        end if;
      end for;
    end if;
  end for;
  return r';
end Candidate;
```

Figure 4.4: Computation of the most convenient route

demand d_{ij} with a route that includes the shortest path between i and j in G. It is desirable that this condition can be fulfilled for a high number of elements in D. Almost every work related to the TNDP agrees that there must exist a route including the shortest path in G between pairs of vertices with high demand. Based on these observations, the algorithm proposed considers the elements of D in decreasing order of demand and generates routes using the shortest path between them.

Since the ideal solution from the users viewpoint is not convenient from the operators viewpoint, the number of routes has to be restricted. For this reason, when we consider the next pair of vertices (u, v) whose demand d_{uv} has to be covered, we test the possibility of including these two vertices in an existing route in the solution under construction. Doing this, we take advantage of the existence of a route, and we modify it so it can serve the demand associated to a pair of vertices which are close to it. Since it is not desirable to extend so much an existing route by inserting vertices in it (because travel time will be increased for demand already served by the route) we impose constraints of maximum route duration and circuity factor to candidate routes resulting from the insertion.

Even if a pair of vertices (u, v) can be inserted in an existing route, we compare the

cost of extending this route with the cost of the shortest path in G between u and v. We always try to minimize the increase of the overall duration of routes in the system. This criterion represents somehow the interest of the operators, since the sum of the durations of all routes in the system is directly proportional to the fleet size (which is an important component in the costs for the operators).

4.4.2 Implementation variants

The PIA constructive algorithm as presented in Figure 4.3 admits some variants in its implementation. The following is a list of modifications that can be considered with different purposes:

1. Concerning the decision of the next element of list l to be considered to cover its corresponding demand, the original strategy (as presented in Figure 4.3) is deterministic, i.e., always the pair of vertices with maximum demand is selected. One possible alternative is to consider a sublist l' of elements with highest demand in l and then select one element from it, in a systematic way. A particular case of this strategy is the greedy randomized construction [45], where a random selection of an element of l' is performed, with a given distribution of probabilities.

2. The shortest path in G is always used when PIA creates a new route to be added to the solution under construction. In [10], the authors observe that by using alternative paths there are chances of covering more demand with a slight increase in the route length; k-shortest paths [122] have been used in [10, 43, 44] in the context of the resolution of the TNDP and this feature can be easily incorporated to the original version of PIA.

3. The proposed algorithm starts with an empty set of routes, but it can be fed with an initial set of given routes R; if so, $D_0(R)$, $D_{01}(R)$ and list l have to be appropriately initialized with information given by R. This characteristic of incremental construction of the algorithm allows for example to consider a set of initial fixed routes given by the planner. Another use of the algorithm can be made when there is a need to complete an unfeasible solution with respect to demand covering constraints, since some algorithms (for example some local search algorithms [57]) manipulate unfeasible solutions at intermediate steps, and may end with no feasible solution.

4. Though PIA uses an undirected graph as underlying model, it can be adapted to work with a directed graph as input network. This constitutes a more realistic modeling since routes may not have the same duration in both ways. Both the structure of routes and some parts of the algorithm (specially those that check and update demand covering) must be modified in order to model this characteristic. The algorithm also can be easily extended to support a graph with different types of nodes (street, stop and centroid).

5. Demand covering with more than one transfer can be considered in the model and implemented in the algorithm. Though it adds more complex subroutines to the algorithm, it may decrease its overall execution time, since under this scenario there

is no need to create or modify routes when a given percentage of the whole demand is already covered with a given number of transfers (therefore the algorithm will stop earlier).

4.5 Experimental study

In this section we perform computational experiments of the proposed algorithm, with the following scope:

1. Comparison between PIA and RGA. This set of experiments was made to compare the behavior of both algorithms, in terms of their sensitivity under changes in demand covering requirements, and in terms of values of Y_1 and Y_2 of the solutions produced, as well as execution times.

2. Analysis of diversity. This second set of experiments was designed to investigate the ability of PIA to produce different solutions, by changing values of some of its parameters. This is made in order to evaluate the usefulness of PIA as a subroutine of a more structured algorithm, which may require a set of diverse solutions.

3. Using PIA to solve the TNDP. This experiment tests the accuracy of PIA, by comparing its results against exact results obtained by the mathematical formulation proposed in Chapter 3. We also obtain a complete solution (routes and frequencies) for a real case using PIA.

The first and second experiments use a real case related to the city of Rivera, Uruguay, which is described in Appendix A. The real case gives a reference for comparison, specially in terms of the required number of routes as a function of demand covering requirements. The third experiment also uses the small cases used in Chapter 3.

Implementations were made in C++; programs were run on a Pentium 4 PC, with a 1.6 GHz processor and 512 MB of RAM. Values of Y_1 are calculated in terms of the elements of the origin-destination matrix, which are expressed in trips per minute; Y_2 is expressed in minutes.

4.5.1 Comparison between PIA and RGA

The implementations of both algorithms use the same data structures and subroutines, thus trying to make the comparison as fair as possible. Both algorithms rely on the availability of pre-computed shortest paths (and their cost) between all pairs of vertices in G. For the implementation of RGA, the criterion of maximum demand per minimum time insertion was implemented [10]. According to this, the vertex which maximizes d_v/t_v is selected (from the set of feasible candidates) for insertion in route r, where d_v is the demand of elements in l between vertex v and vertices on r, and t_v is the cost increase of the route resulting after the insertion of v. The implemented procedure **ExpandRoute** applies three out of the six constraints of the original version of RGA, namely, loop avoiding, maximum route duration and maximum circuity factor.

Sensitivity for different levels of demand covering

As mentioned in [10], increasing the imposed levels of demand covering causes an increase in the number of routes and total route mileage in the solutions produced by RGA. This tendency was also expected in PIA, but to a lower extent.

In this experiment we run both algorithms for different combinations of D_0^{min} and D_{01}^{min}, and we investigate its effect in values of deviation from the shortest path Y_1, total route duration Y_2 and number of routes $|R|$.

Table 4.2 shows that parameter D_{01}^{min} is the main factor that rules the increase of Y_2 and $|R|$ for both algorithms; values of Y_1 do not necessarily increase because they are summations of ratios that can decrease (new routes are needed in order to cover more demand, which can decrease the travel time of the already covered demand), in particular for higher levels of D_0^{min}. While increasing D_0^{min} for a fixed value of D_{01}^{min} does not impact so much (values along the same column), increasing D_{01}^{min} has a significant impact (values along the same row).

When comparing PIA with RGA, we can observe that while values of Y_1 vary in a similar way, values of Y_2 and $|R|$ increase to a high extent for RGA. For example, for a fixed value of $D_0^{min} = 0.50$, an increase of D_{01}^{min} from 0.50 to 1.00 causes an increase in Y_2 of 427% for PIA and 728% for RGA; the increase in $|R|$ is 467% for PIA and 767% for RGA. This shows that when we increase the amount of demand covered to that level (which is desirable from the users viewpoint), the cost increase for the operators (represented by number of routes and total route duration) when using RGA is around 65% higher than the cost increase using PIA. The cost increase for the users (represented by travel time) is almost the same for both algorithms.

Objective values of the produced solutions

In this experiment we compare the results produced by both algorithms in terms of functions Y_1 and Y_2, which are intended to be minimized when solving the TNDP. We also compare number of routes and execution time values.

In order to compare, we implemented the greedy randomized construction variant explained in Section 4.4.2 (implementation variant number 1), in both PIA and RGA. The list l' is constructed by selecting the $\alpha|l|$ elements with highest demand of l, where $\alpha \in [0, 1]$ is a parameter. The random selection of an element (u, v) from l' is made by using a biased probability distribution [104], where $bias(u, v) = d_{uv}$, and the corresponding probability is

$$\text{Prob}(u, v) = \frac{bias(u, v)}{\sum_{(i,j)\in l'} bias(i, j)} \ . \tag{4.5}$$

This means that the probability of choosing the pair (u, v) is proportional to its demand.

Demand covering related parameters were set as $D_0^{min} = D_{01}^{min} = 1.00$; we impose this strong requirement of demand covering for two reasons: (i) to compare results under extreme conditions of required demand covering, (ii) to obtain solutions which are comparable to the public transportation system of Rivera, where all the demand is served without transfers. Values of parameters over routes were $\rho_{max} = 1.5$ and $t_{max} = 120$ (minutes). Parameter α for the greedy randomized construction was set to 0.2.

Table 4.2: Sensitivity under changes in levels of required demand covering

(a) PIA

D_{01}^{min}	0.50	0.75	0.95	0.99	1.00
D_0^{min}			Y_1		
0.50	11.35	12.61	15.90	16.20	16.09
0.75	-	14.41	15.90	16.20	16.09
0.95	-	-	16.24	16.20	16.09
0.99	-	-	-	16.14	16.09
1.00	-	-	-	-	16.09
D_0^{min}			Y_2		
0.50	207.48	306.63	713.24	831.40	1093.04
0.75	-	359.68	713.24	831.40	1093.04
0.95	-	-	763.09	831.40	1093.04
0.99	-	-	-	1042.43	1093.04
1.00	-	-	-	-	1117.98
D_0^{min}			$\|R\|$		
0.50	3	4	10	11	17
0.75	-	5	10	11	17
0.95	-	-	10	11	17
0.99	-	-	-	15	17
1.00	-	-	-	-	18

(b) RGA

D_{01}^{min}	0.50	0.75	0.95	0.99	1.00
D_0^{min}			Y_1		
0.50	12.20	13.46	16.17	16.26	15.97
0.75	-	15.16	16.17	16.26	15.97
0.95	-	-	16.01	16.26	15.97
0.99	-	-	-	16.17	15.97
1.00	-	-	-	-	15.97
D_0^{min}			Y_2		
0.50	280.14	368.47	1106.24	1602.67	2319.86
0.75	-	717.63	1106.24	1602.67	2319.86
0.95	-	-	1486.28	1602.67	2319.86
0.99	-	-	-	1991.08	2319.86
1.00	-	-	-	-	2402.13
D_0^{min}			$\|R\|$		
0.50	3	4	12	18	26
0.75	-	8	12	18	26
0.95	-	-	16	18	26
0.99	-	-	-	22	26
1.00	-	-	-	-	27

	Execution	Y_1	Y_2	$\|R\|$	*over*	T
	1	16.20	1179.83	19	660.45	15.20
	2	16.32	1198.66	19	665.79	15.16
	3	16.75	1098.18	19	584.33	13.39
	4	16.09	1212.77	19	662.04	13.77
	5	15.90	1190.65	19	647.47	13.11
PIA	6	16.46	1225.81	16	697.35	15.23
	7	16.14	1074.17	19	560.99	11.56
	8	16.65	1128.35	15	645.15	12.23
	9	15.65	1218.93	20	673.14	15.42
	10	17.04	1081.83	18	578.34	11.25
	Average	16.32	1160.92	18	637.51	13.63
	1	16.30	2114.59	23	1503.43	0.48
	2	16.10	1851.23	20	1250.00	0.39
	3	15.89	1960.08	21	1338.90	0.38
	4	16.49	1997.97	22	1392.33	0.52
	5	16.16	1818.06	20	1210.66	0.39
RGA	6	16.13	2307.79	24	1681.13	0.44
	7	16.05	2040.36	22	1395.03	0.34
	8	16.10	1966.98	21	1358.40	0.39
	9	15.71	2144.98	23	1527.34	0.41
	10	15.92	2036.59	21	1403.43	0.33
	Average	16.09	2023.86	22	1406.07	0.41

Table 4.3: Results of 10 independent executions

Table 4.3 shows for each set of routes R produced by each algorithm in 10 independent executions, values of functions Y_1 and Y_2, number of routes $|R|$ and execution time T (in seconds). We also calculate a measure of the overlapping of routes in a solution, defined as

$$over(R) = \sum_{e \in E} c_e \max\{0, R_e - 1\} \ ,$$

where R_e is the number of routes in R which use edge e.

We can observe that while both algorithms produce results with similar values in Y_1, when comparing averages, RGA produces higher values than PIA in number of routes (22% higher) and total route duration (74% higher). This difference is accompanied by an overlapping of routes which is more than 100% higher for RGA, suggesting that there are many routes serving the same demand in solutions produced by this algorithm. On the other hand, execution time is highly favorable for RGA, which is on average 27 times lower than the execution time for PIA. This significant difference can be explained by (i) the quadratic computational complexity of the **Candidate** subroutine in PIA, and (ii) the computational burden resulting from the handling of paths in the insertion of pairs of vertices in PIA.

Table 4.4 shows results from 1000 independent executions of both algorithms, summarized in minimum, average and maximum values. We can observe that averages remain approximately the same as those in Table 4.3. Minima in Y_1 are very similar between PIA and RGA, and relatively close to its lower (theoretical) possible value, that is attained when $t_{ij}(R)/t^*_{ij} = 1$ for every (i, j), in this case, 13.94. One remarkable fact is that PIA

		Y_1	Y_2	$\|R\|$	*over*	T
PIA	Minimum	15.26	903.60	12	401.74	6.91
	Average	16.30	1146.08	18	615.23	12.82
	Maximum	17.67	1441.05	24	860.55	21.36
RGA	Minimum	15.49	1429.40	15	880.49	0.27
	Average	16.18	1998.44	22	1377.50	0.47
	Maximum	17.47	2610.48	28	1950.24	1.13

Table 4.4: Summarized results of 1000 independent executions

has produced a solution with 12 routes, less than the number of routes in the real system of Rivera, which is 13; on the other hand, the smallest set of routes produced by RGA has 15 elements

4.5.2 Analysis of diversity

Since PIA can be used as a subroutine of another algorithm (for example, a metaheuristic working with populations or with a multi-start strategy), it can be useful to obtain different (diverse) solutions; this type of diversity is considered with respect to decision variables. On the other hand, given the multi-objective nature of the TNDP, it may be desirable to obtain solutions with different trade-off levels between the conflicting objectives of users and operators; this type of diversity is considered with respect to the objective space.

In this section we study the diversity in decision space by computing a measure of similarity among solutions, which takes into account the structure of their routes (in terms of the sequences of vertices). Diversity in objective space is grasped graphically, by plotting the results in a two-dimensional space defined by the objectives.

Diversity in decision space

When using PIA as a subroutine of an approximative algorithm for the TNDP, a possible requirement imposed to it, is the ability to generate a diverse set of solutions (sets of routes). Some metaheuristics require a constructive algorithm capable of producing several solutions, being each one different from the other ones (see for example GRASP, Genetic Algorithms and Scatter Search in [56]). This difference (or diversity) is considered with respect to decision variables, in this case the structure of routes. Several existing metaheuristic based algorithms for the TNDP have this requirement on the constructive algorithm [94, 103, 113].

In this work, we propose a diversity measure *diver* over a set of solutions $\Re = \{R_1, \ldots, R_m\}$, which is defined as

$$diver(\Re) = 1 - \frac{\sum_{(R_i,R_j)\in\Re, i<j} sim(R_i, R_j)}{|\Re|(|\Re|-1)/2} ,$$

where *sim* is a measure of similarity between two sets of routes calculated as

$$sim(R_i, R_j) = \frac{\sum_{r\in R_i} sim(r, R_j) + \sum_{r\in R_j} sim(r, R_i)}{|R_i| + |R_j|} .$$

α/k	1	2	5	10	20	50
deterministic	0.00	0.00	0.00	0.00	0.00	0.00
0.2	0.31	0.37	0.39	0.44	0.44	0.45
0.4	0.30	0.37	0.42	0.43	0.44	0.46
0.6	0.31	0.38	0.41	0.42	0.46	0.47
0.8	0.32	0.37	0.41	0.43	0.45	0.47
1.0	0.34	0.36	0.41	0.44	0.44	0.46

Table 4.5: Diversity in decision space

Similarity between sets of routes is calculated in terms of the similarity of a route with respect to a set of routes and it is defined as

$$sim(r, R) = \max_{r_i \in R} \{sim(r, r_i)\} ,$$

where $sim(r, r_i)$ is the proportion of the cost of the arcs in r which are already included in r_i with respect to its cost, i.e. $sim(r, r_i) = \sum_{e \in r \wedge e \in r_i} c_e / \sum_{e \in r} c_e$. Thus, the diversity measure is based on a similarity measure that takes into account the structure of routes in detail. Note that $diver(\Re) \in [0, 1]$, where 0 states that all solutions in $\Re$ are identical (meaning that $\Re$ is not a diverse set) while 1 indicates the opposite situation (i.e. any pair of sets of routes are built over totally different sets of edges).

In order to obtain diverse solutions in decision space by using PIA, we change its α parameter and we use the implementation variant number 2 (explained in Section 4.4.2), where the new route to be considered for addition to the solution under construction is generated by using one of the k-shortest paths (k is a parameter); the selection is made randomly with a uniform probability distribution in the discrete interval $[1..k]$. The k-shortest paths are generated in a previous step using Yen's algorithm [122].

In this experiment we vary α and k, and for each combination of these parameters we perform 10 independent executions of PIA, thus generating sets of 10 solutions over which diversity values are computed and shown in Table 4.5. Observe that *diver* is not so sensitive to parameter α; any value of $\alpha > 0$ will cause PIA to produce a diverse set of solutions (since $\alpha = 0$ represents the deterministic version of the algorithm). Parameter k shows a higher influence in diversity, with a monotonically increasing tendency; diversity increases even for high values of k, however, we must take into account that respective solutions are not necessarily good in terms of Y_1 and Y_2.

We observe that diversity is always far from 1.00. This fact suggest that still it may be possible to improve the diversity of the results, by introducing other mechanisms in the algorithm; however, we must consider that the fulfilment of demand covering constraints (specially for higher values of D_0^{min}) keeps the maximum reachable diversity bounded.

Diversity in objective space

Since the TNDP is a multi-objective problem, where objectives of users and operators are conflictive [65], a possible use of PIA consists in producing solutions with different trade-off levels between these objectives (in our case represented by functions Y_1 and Y_2 respectively). These different solutions can be obtained simply by taking advantage of the stochastic nature of the greedy randomized version of the algorithm. However, by

changing the parameter of maximum duration of routes t_{max}, a wider range of trade-off levels can be obtained. In this experiment we made 10 executions of the greedy randomized version of PIA with the same parameter configuration as that used in Section 4.5.1; at each execution the value of t_{max} is randomly selected (with uniform probability) from the real interval $I = [40, 120]$ (minutes).

Several measures have been proposed to evaluate the diversity in objective space of a set of solutions for a multi-objective problem [31]. However, all of them must be applied to a non-dominated set of solutions. Because we are interested in getting an idea of the diversity in objective space of all obtained solutions, we use a graphical method consisting in plotting the solutions in the two-dimensional space defined by Y_1 and Y_2.

Figure 4.5 shows the 10 solutions obtained by the greedy randomized version of PIA (set $\Re_1$, already shown on Table 4.3) as well as the 10 solutions obtained by the same version of the algorithm also including the variation of t_{max} as explained before (set $\Re_2$). We can see that there is a region of the objective space (corresponding to low values of Y_1 and high ones of Y_2) that is covered by $\Re_2$ and not covered by $\Re_1$. This shows that by changing the parameter t_{max}, the algorithm is able to produce solutions with very low cost for the users (and very high for the operators), which can not be obtained by the greedy randomized version. Despite the fact that these solutions may represent an extreme trade-off level between the conflicting objectives (and therefore they may not be practicable in the real system), they may be useful in the context of an algorithm that is designed to produce a Pareto front as solution of the multi-objective TNDP. The highest value reached by PIA for Y_2 in 1000 independent executions using only the greedy randomized version is 1,441.05 (see Table 4.4), which is significantly lower than the maximum reached in $\Re_2$ (which is greater than 2,100.00, with a low value in Y_1, suggesting that it is close to the optimal Pareto front [31]). Thus, the variation of parameter t_{max} allows the algorithm to produce solutions in a wide range of trade-off levels between functions Y_1 and Y_2. It is interesting to note that diversity in objective space does not imply necessarily diversity in decision space for solutions in $\Re_2$; we observe that $diver(\Re_2) = 0.27$, which is lower than the minimum diversity already shown in Table 4.5.

4.5.3 Using PIA to solve the TNDP

The goal of these experiments is to compare results produced by PIA against optimal solutions for the TNDP. For doing that, we use PIA as a subroutine of a 2-phase heuristic that solves the problem stated by formulation OPT2 (Section 3.2.3); thus we are under conditions to compare the results with those obtained by using directly the formulation with a MILP solver (Section 3.5). Since we want to evaluate the accuracy of PIA but it only determines the routes of a solution, we then determine the optimal frequencies for those routes by applying OPT2. By comparing that solution with the optimal one we are somehow evaluating the contribution of PIA in obtaining a complete solution to the TNDP; note that the comparison is likely to be independent on the way in which the frequencies are determined, sice they are optimal for those routes. The 2-phase approach explained above is similar to several existing heuristic approaches that determine the routes in a first stage and the frequencies in a second one [9, 77, 103]. We also use the 2-phase heuristic to generate a complete solution (routes and frequencies) for the case of Rivera (Appendix A).

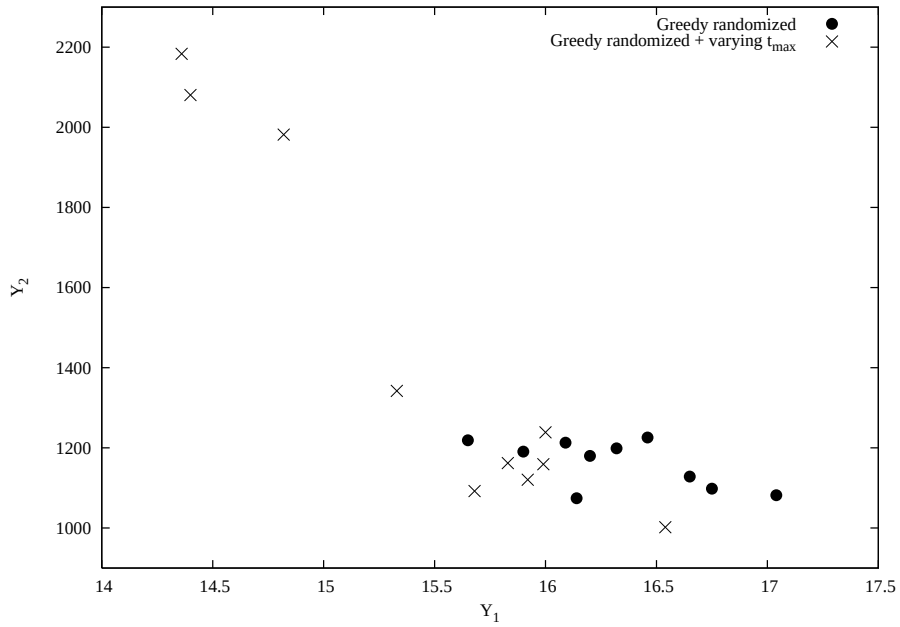


Figure 4.5: Diversity in objective space

Comparison between exact and approximated results

We use the 2-phase heuristic to obtain approximate solutions corresponding to the instances considered in Section 3.5.1. The parameter setting used for PIA was $t_{max} = 200$ for Small and $t_{max} = 1.2$ for Wan and Lo, while $\rho_{max} = 1.5$ for both cases. Table 4.6 shows the objective values already shown in Table 3.2 (column *Opt*), the approximated value obtained by the 2-phase heuristic (column *Heu*) and the percentage gap between these two values, relative to the first one. We can observe that gaps are relatively constant for the case Small, while for the case of Wan and Lo the gap is relatively low. We should remember that the optimal value for this case was computed using a subset of all possible routes and a time limit (Section 3.5.1), therefore some routes (that could be found by PIA) may be missed and some parts of the feasible space could remained unexplored in that computation.

We note that we are not proposing an approximate algorithm to solve the TNDP; we only wanted to have an idea of closeness to the optimum of results obtained by PIA. The 2-phase heuristic presented here is very simple, so one can expect that the approximation can be improved by using more sophisticated ideas. Moreover, the small sizes of the test cases used do not allow PIA to perform all of its logical branches; therefore these tests can not be taken as a strong evaluation of the accuracy of the heuristic. Part of the motivation of the experiment presented in the next section is given by this fact.

Obtaining a solution for Rivera

The 2-phase heuristic was applied to the case of the city of Rivera. We used the following parameter setting, whose values are suggested by the solution (routes and fre-

Instance	*Opt*	*Heu*	*%Gap*
Small $B = 8$	385.00	460.00	19
Small $B = 20$	291.50	355.00	22
Small $B = 60$	254.20	292.50	15
Wan and Lo	1778.19	1807.50	2

Table 4.6: Comparison of exact and approximated results

quencies) of the public transportation system of Rivera: $t_{max} = 80$, $\rho_{max} = 1.5$, $\Theta = \{1/60, 1/40, 1/30, 1/20\}$ and $B = 25.65$. Note that it is not possible to obtain the optimal value using the mathematical formulation OPT2, since the size of the set of all different routes in G is not manageable. In order to provide a reference of distance to optimality, we compute lower bounds for both on-board and waiting time (terms of objective function (3.22)), using the definitions presented in Section 5.5.1 (tv^* and tw^* respectively). The solution found attained an objective value of 604.79, comprising 16 routes with an average route duration and route headway of 74 and 52 minutes respectively. The distance to on-board travel time lower bound tv^* is 21% while the distance to waiting time lower bound tw^* is 31%. These bounds are provided only in order to give an idea of distance to optimality, despite they may be weak; for example, they do not take into account the value of the fleet size constraint, and the one related to waiting time is actually a pseudo lower bound (Section 5.5.1). Objective value, number of routes, average route duration and average route headway are 9%, 12%, 17% and 41% higher than their respective values corresponding to the solution of Rivera. Note that even though our frequencies are optimal for a given set of routes (obtained from a single deterministic execution of PIA), they are very low in comparison with the ones of Rivera; the difference is that the frequencies of the lines of Rivera are defined over a set of routes that is "optimized" by the planners of the municipality. We resume the comparisons with the solution operating the public transportation system of Rivera in Section 5.5.4.

4.6 Conclusions and future work

The PIA algorithm proposed modifies the RGA by using a new strategy of insertion of pairs of vertices, instead of the original expansion of routes by inserting single vertices.

When compared with RGA, PIA produces solutions with similar values of on-board travel time, and significantly better in terms of number of routes and total route duration. On the other hand, execution time is significantly higher; this fact is explained mainly by the quadratic computational complexity of the subroutine of insertion of pairs of vertices, which is intensively used by PIA. Further investigation looking at possible strategies to reduce the complexity of the algorithm are required to improve execution times. However it is worth mentioning that execution time is not the main concern in the context of strategic planning, where TNDP takes place. Though execution times are highly favorable to RGA, the cost of the solutions produced by PIA from the operators viewpoint are much lower, while the cost for the users is almost the same. In terms of the real transit system, this reduction of operation costs may imply a reduction on fares and/or subsidies, while maintaining the revenues of the operators and the level of service for the users.

The algorithm has shown to be flexible to be used as a subroutine in other algorithms such as metaheuristics; it is capable to produce diverse solutions in both decision and objective spaces. It is used in [90] to generate solutions with different trade-off levels between the objectives of users and operators, in the context of a metaheuristic based algorithm; it is also used in [88] to complete unfeasible solutions with respect to demand covering constraints.

A real medium to small-sized test case was used in this work. The algorithm has shown to be capable of producing solutions which are comparable (in terms of number of routes) to real solutions. For cases of much larger sizes, the applicability of the algorithm has to be tested, specially with regard to execution time. Demand covering plays an important role on those cases: high imposed levels of demand covering without transfers may impact on the performance of the algorithm, by increasing its execution time. Also the possibility of covering the demand with more than one transfer has to be considered in both model and algorithm for big cases, since the budget of the operators will not allow direct connections as users might wish; this impacts on the complexity of the implementation of the algorithm but not necessarily it degrades its computational performance (as explained in Section 4.4.2).

We use PIA as a subroutine of a simple heuristic that obtains a complete solution (routes and frequencies) for the TNDP; its results are compared with exact results computed by using a mathematical programming formulation. Thus, we provide a reference to evaluate the accuracy of PIA. It would be also interesting to study how close are the solutions produced by PIA to Pareto optimal solutions according to objectives Y_1 and Y_2.

Chapter 5

Multi-objective metaheuristic approach to route optimization[‡]

In this chapter we present a multi-objective metaheuristic approach for the TNDP. The motivations for the research related to this part of the thesis are:

- The TNDP is a complex combinatorial problem, therefore a heuristic approach seems to be a suitable alternative to solve real instances.

- The problem has an intrinsic multi-objective nature, given that the objectives of users and operators are conflictive. Roughly speaking, this mean that an improvement in one objective is attained only with a detriment on the other one.

- The multi-objective nature of the TNDP has been treated in the literature by reducing the problem to a single-objective one, either by weighting both objectives into a single objective function or by considering one objective as a constraint. These methods arrive to a single solution that highly depends on the parameters used (weights in the objective function and values in the constraint), which may be difficult to set for the planner.

- In case we want to obtain a set of solutions representing different levels of trade-off between the conflicting objectives, a straightforward approach is to run a single-objective algorithm repeatedly with different weighting or constraining parameters. A heuristic algorithm that exploits the search to obtain solutions with different levels of trade-off in a single run seems to be a more efficient approach.

- The only previous work that applies a multi-objetive approach to the TNDP is [65], where the authors solve the multi-objective problem by using a special purpose heuristic algorithm that obtains a small set of non-dominated solutions. By contrast, multi-objective metaheuristics is a promising technique to be applied to the TNDP in the context of a multi-objective approach to the problem, in the sense that more non-dominated solutions could be found.

[‡] Different parts of the content of this chapter were published in [90] and presented in [87] and [89].

- It is desirable to test the multi-objective approach with a real case; a comparison of the solutions obtained against the solution of the real system would provide elements to validate the approach.

In this work we model the TNDP as a multi-objective combinatorial optimization problem and we propose an algorithm based on the GRASP metaheuristic to solve it; as a multi-objective metaheuristic, the algorithm produces in a single run a set of non-dominated solutions representing different trade-off levels between the conflicting objectives of users and operators. The case proposed by Mandl is used to show that the multi-objective metaheuristic is capable of producing a diverse set of solutions, which are compared with solutions obtained by other authors. We show that the proposed algorithm produces more non-dominated solutions than the Weighted Sum Method with the same computational effort, using the cases of Mandl and the city of Rivera. We also show that the proposed algorithm produces solutions which are comparable with the solution of the real system of Rivera.

5.1 Introduction

The main objective in the design of routes for a public transportation system is the maximization of the level of service offered to the users [33], subject to constraints on infrastructure, policy and budget. One of the most important budgetary constraint is related with the operation cost of the services. In a general sense, a better service is offered to the users in the presence of more routes and high frequencies. But there exist upper bounds for the resources of the service provider that make their operation profitable (fares and subsidies are also bounded). Then a convenient trade-off level has to be established, maybe entailing the evaluation of various alternative system designs. Thus, the problem of the optimal design of routes and frequencies has an intrinsic multi-objective nature.

The exact resolution of the TNDP has the following difficulties, enumerated among others in [9, 20]:

- High combinatorial complexity: [65] classified the problem as a complex variant of the generalized network design problem [76], which is NP-hard.

- The TNDP requires an assignment submodel: the evaluation of a given solution (from the users viewpoint) needs a behavior model of the passengers concerning the routes and frequencies of the solution.

- Multi-objective nature: the existence of conflicting objectives adds complexity to the problem, either in the *a priori* estimation of the relative importance of the objectives, or in the calculation of several solutions with different trade-off levels between the conflicting objectives.

The combinatorial complexity of the TNDP has been tackled in the existing literature almost exclusively by means of inexact methods. Complete enumeration of feasible solutions is prohibitively expensive; mathematical programming formulations exist only for simplified versions of the problem [15, 107, 120]. The first algorithms published for

the TNDP were heuristics [7, 10, 77, 109]. Lately, several applications of metaheuristics have been proposed, most of them using Genetic Algorithms with different coding schemes [94, 103, 113], Tabu Search [44] and Simulated Annealing [43].

The assignment models generally used in the context of the TNDP aim to give a realistic representation of the interaction between passengers and buses; but their complexity must be kept bounded given the impact they have in the overall efficiency of the optimization algorithms. The most used approaches are all-or-nothing assignment [94], common lines and transfers [9], and detailed network treatment [41].

Multi-objective optimization problems require a different treatment than single-objective ones. Instead of having a single optimal solution, they have a set of different non-dominated solutions which represent different trade-off levels among the conflicting objectives. In order to identify a single solution from that set, additional information (usually provided by a decision maker, in our terminology, the planner) is required, concerning the relative importance of the conflicting objectives. There are several ways to take into account this information in the overall multi-objective optimization process [37], namely: (i) in the *a priori mode* the preferences among different objectives are known at the beginning of the process and the optimization technique uses this information to find an optimal solution, (ii) in the *a posteriori mode* the information from the decision maker is used to analyze a set of non-dominated solutions previously generated, (iii) in the *interactive mode* preferences are introduced during the process, which alternates computing steps with preference setting, requiring a high participation level of the decision maker.

Most of the previous work on the TNDP have considered the multi-objective nature of the TNDP by using an *a priori* estimation of a vector of weights to express a particular trade-off level between the conflicting objectives [9, 41, 94, 103, 113]. The *a posteriori* mode has been adopted only in [65] and [86]. To the best of our knowledge no *interactive* multi-objective optimization method has been applied to the TNDP

All the existing metaheuristic based algorithms for the TNDP, with the exception of [86], solve a single-objective optimization problem by summarizing the different objectives into a single one by using a vector of weights. In recent years, a growing amount of work has been published about metaheuristics specially designed for multi-objective combinatorial optimization problems. This type of algorithms has been denominated as *multi-objective metaheuristics* and they are defined by different authors as: methods that aim at generating a good set of non-dominated solutions in a single run [66], and algorithms that deal with the multiple objectives directly [38]. The basic idea of multi-objective metaheuristics is to adapt the mechanisms of their original single-objective counterparts to handle effectively and efficiently multi-objective optimization problems [24, 36, 38].

In this work, we present a heuristic based on the GRASP metaheuristic [45, 104] to solve the TNDP with a multi-objective approach (i.e., adopting the *a posteriori* mode). It allows to obtain in a single run, a set of non-dominated solutions representing different trade-off levels between the conflicting objectives. A previous GRASP based algorithm was developed by the author of this thesis [86]; its construction and local search components are completely different. The multi-objective algorithm proposed uses the Pair Insertion Algorithm (Chapter 4) to construct a set of routes, and a neighborhood definition that is used to search for a near optimal set of frequencies for a particular trade-off level. The assignment model of Baaj and Mahmassani [8] is used to distribute the demand among

the routes of a given solution and to calculate some variables needed by the optimization procedure. The proposed methodology is tested with the benchmark test case proposed by Mandl [9] and the case relative to the city of Rivera, Uruguay (Appendix A). Existing and proposed measures are calculated for different non-dominated solutions in order to show the ability of the proposed algorithm to generate a diverse set of solutions; these measures may be useful for the decision maker. The obtained solutions are compared (in terms of objective values) with solutions published in the literature and with the solution of the real system of Rivera. Moreover, we show that the multi-objective metaheuristic algorithm produces more non-dominated solutions than its single-objective version used as a subroutine in the Weighted Sum Method [31], with the same computational effort.

The remaining of the chapter is organized as follows. We formally state the problem and used notation in Section 5.2 and present the adopted multi-objective approach in Section 5.3. Details of the GRASP implementation are given in Section 5.4. Numerical results are presented in Section 5.5, and finally some conclusions and future work are formulated in Section 5.6.

5.2 Problem definition and notation

Our model is inspired in the work of Baaj and Mahmassani [9]. The main reason for using an existing model is that we want to compare our results with existing results published in the literature; this requires specially to use the same assignment model, in order to compare objective values under the same hypothesis about the behavior of the passengers. We use this particular model because it is the only one published in the literature which contains a detailed description of the assignment sub-model and data to validate our implementation. Also this model is consistent with the hypothesis stated in Section 1.1.

As we do in Chapter 4, we assume that an infrastructure graph $G = (V, E)$ and an origin-destination matrix D (defined in Chapter 2) are given. We adopt the simplified model explained in Section 2.1.3 for representing the routes and we assume that all vertices are of type street, stop and centroid at the same time. A solution S to our problem is a pair (R, F) where $R = \{r_1, \ldots, r_r\}$ is the set of routes and $F = \{f_1, \ldots, f_r\}$ is the set of their corresponding frequencies; each f_k is a positive real value that represents the inverse of the average time between subsequent vehicles on route r_k. We denominate line k to the pair (r_k, f_k). Given a solution S, the assignment model produces the corresponding flows v over the trajectory graph (Section 2.2). Let $\Phi_k = \{v_a\}$ be the set of flows in travel arcs a that belong to route r_k, computed by the assignment model.

The conflicting objectives of users and operators are modeled with functions Z_1 and Z_2, respectively, which have to be minimized simultaneously. The first function,

$$Z_1(S) = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} d_{ij}(tv_{ij} + tw_{ij} + tt_{ij}) \tag{5.1}$$

expresses the overall time needed to transport the users between their corresponding origin and destination vertices. It has three components: on-board travel time tv, waiting time tw and transfer time tt. These values are determined by the assignment model; tv_{ij} is calculated using the costs of the edges of G that are used by lines in S connecting vertices

i and j; tw_{ij} depends on the frequencies of these same lines; tt_{ij} is a penalty (expressed in time units) which represents the discomfort of transfers from the users viewpoint (we define σ_t as the penalty of each demand unit which has to perform transfers).

The objective of the operators is represented by the fleet size (Section 2.3),

$$Z_2(S) = \sum_{r_k \in R} f_k t_k \ , \tag{5.2}$$

where $t_k = 2\sum_{e \in r_k} c_e$ is the total duration (round-trip time) of route r_k.

For a given solution $S = (R, F)$, we consider demand covering constraints as defined in Section 4.2, namely

$$D_0(S) \geq D_0^{min} \ , \tag{5.3}$$

$$D_{01}(S) \geq D_{01}^{min} \ . \tag{5.4}$$

We consider lower and upper values for frequencies, f_{min} and f_{max}, respectively. While the former takes care of the level of service offered to the users, the latter represents a limit imposed by the operational possibilities of the transit mode. These constraints are expressed as

$$f_{min} \leq f_k \leq f_{max} \qquad \forall f_k \in F \ . \tag{5.5}$$

The maximum load factor constraint imposes an additional condition for the frequencies. It is expressed as

$$f_k \geq \frac{\phi_k^*}{\eta\omega} \qquad \forall f_k \in F \ , \tag{5.6}$$

where $\phi_k^* = \max\ \Phi_k$ is the critical flow in route r_k and ω is the seating capacity of vehicles. The given constant $\eta \geq 1$ is the maximum load factor in vehicles, expressing a tolerance in the number of standing passengers. According to this, $\eta\omega$ is the maximum allowed capacity of vehicles.

Note that although the formulation of this model for the TNDP is not explicit, we can identify an underlying bilevel structure consistent with the one stated by formulation OPT3 (Section 3.4.2). Thus, while constraint (5.5) and objective function (5.2) depends on variables of the upper level only (routes and frequencies), all the other constraints and objective function (5.1) depends on values (flows and travel times) that are computed by the assignment submodel. In particular, (5.3)-(5.4) represent the transfer constraint (3.60) for $\tau = 1$ and (5.6) represents the bus capacity constraint (3.61) of OPT3.

We denominate as $\mathcal{P}$ the problem defined by the simultaneous optimization of objective functions (5.1) and (5.2), under constraints (5.3)-(5.6).

5.3 Multi-objective approach

Given the multi-objective nature of problem $\mathcal{P}$, it does not have a single optimal solution S^*; instead it has a set of *Pareto optimal solutions* P^*, called *optimal Pareto front* [31]. The optimal Pareto front of a multi-objective optimization problem is the non-dominated set of the whole set C of feasible solutions. The non-dominated set of a given set C is made up of all the solutions that are not dominated by another solution in C. A solution S_1 dominates another solution S_2 if S_1 is no worse than S_2 in all objectives and S_1 is

strictly better than S_2 in at least one objective. If any of these two conditions is not true, then S_2 is not dominated by S_1. When we refer to elements in the feasible set C, we are dealing with the *decision space*, i.e., the space where variables take values. On the other hand, domination is defined according to the values of the objective functions evaluated over solutions of C in the *objective space*. In the context of the TNDP, the decision space is made up of all sets of routes with frequencies (and corresponding demand assignment) satisfying constraints (5.3)-(5.6). The objective space is a two-dimensional space defined by functions Z_1 and Z_2.

Problem $\mathcal{P}$ can be classified according to [36] as a multi-objective combinatorial optimization (MOCO) problem. The discrete nature of the variables that represent the structure of routes gives the combinatorial characteristic. The conflicting objectives represented by functions (5.1) and (5.2) result in the existence of a set of Pareto optimal solutions instead of a single optimal solution. Note that other conflicting objectives could be considered, such as those related to land use and emissions. However, additional (sub)models and therefore additional data will be required in that case.

The multi-objective nature of the TNDP as it is posed by objective functions (5.1) and (5.2) (or by similar formulations), has been tackled by the following approaches in the existing literature:

- Making an *a priori* estimation of the relative importance of the conflicting objectives in the form of a vector of weights, and then solving a single-objective optimization problem [94, 103, 113]. Some authors suggest that by varying the values of these weights, a set of solutions with different trade-off levels can be obtained [9].

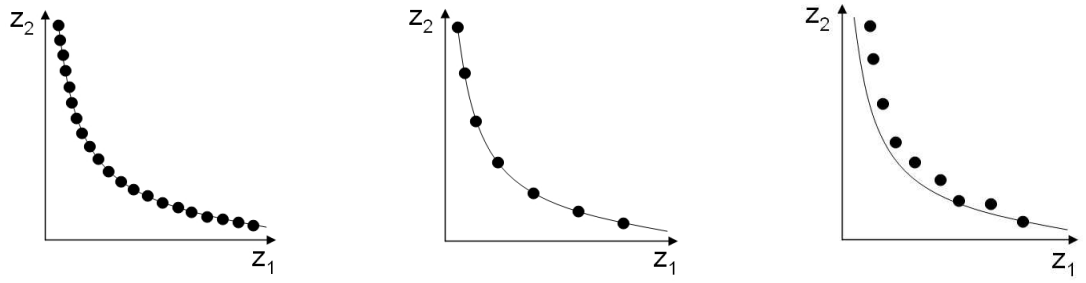
- Calculating a set of non-dominated solutions and selecting *a posteriori* a single one [65, 86].

According to [37], the interactive multi-objective optimization method is the most used in solving practical problems. However, to the best of our knowledge it has not been published any application of this method to the TNDP.

A priori setting of weights requires estimating coefficients whose role in the optimization model is twofold, namely: (i) expressing the relative importance between the objectives, and (ii) conversion between different units of the objective functions Z_1 and Z_2 (time and buses, respectively). No explicit method to set these coefficients is given in the literature related to the TNDP. In this work we adopt the *a posteriori mode* [37], that we call multi-objective approach for the TNDP. We propose an algorithm to find a set of non-dominated solutions which can be used in subsequent steps by the decision maker, either to select a single non-dominated solution, to compare the solutions with an existing solution or to evaluate alternative solutions.

5.3.1 Multi-objective metaheuristics

The exact solution of $\mathcal{P}$ involves finding all its Pareto optimal solutions [37] (Figure 5.1(a)). Most MOCO problems are proven to be NP-hard [53] as well as #P-hard [115] (this is true even for problems which have efficient algorithms in the single-objective case) [38]. This implies that it is unlikely that a polynomial time algorithm exists to exactly solve them, and even to count the elements of the optimal Pareto front. Also from a practical



(a) Optimal Pareto front (b) Pareto optimal solutions (c) Approx. Pareto front

Figure 5.1: Different approaches to solve the multi-objective TNDP

viewpoint, it may not be convenient to bring all the Pareto optimal solutions to the decision-maker. In [105], a method to overcome this difficulty in bi-objective discrete optimization problems is presented, allowing to obtain a subset of the entire Pareto front (Figure 5.1(b)). However, it requires an exact method for solving the associated single-objective optimization problem.

In this work we present an algorithm that is approximate in two senses: (i) it produces a set of non-dominated solutions representing different trade-off levels between the objective functions (however, other non-dominated solutions may be missing), (ii) there is no guarantee of Pareto optimality for each solution found. The algorithm is a multi-objective metaheuristic; therefore, it produces in a single run an *approximate Pareto front* for $\mathcal{P}$ (Figure 5.1(c)).

The main algorithmic difference of multi-objective metaheuristics with respect to their single-objective counterparts is the adopted search mechanism to obtain an approximate Pareto front. In multi-objective optimization, the goals of an inexact algorithm are two, namely [31]: (i) closeness: one seeks to find solutions which are close to the optimal Pareto front, (ii) diversity: one also wants to obtain a set of non-dominated solutions which covers different areas of the objective space, representing a diverse set of trade-off levels between the conflicting objectives. On seeking closeness and diversity, specially designed search mechanisms are needed, which have to deal with both decision and objective spaces. Surveys on this topic can be found in [24, 38].

5.3.2 Multi-objective GRASP for the TNDP

GRASP (Greedy Randomized Adaptive Search Procedure [45, 104]) is a metaheuristic for combinatorial optimization problems, consisting on the repeated execution of a solution construction procedure followed by a local search. The construction is performed using a greedy criterion, by adding iteratively to a solution, elements which are randomly selected from a candidate list. The local search requires the definition of a neighborhood structure, through which to successively advance in the direction of improvement of the objective function. The sequence of construction and local search (GRASP iteration) is repeated a given number of times, obtaining different trajectories in the feasible space. Finally, the best found solution is returned.

In this work we adapt GRASP to solve the TNDP with a multi-objective approach.

Existing adaptations of GRASP for MOCO problems can be found in [11, 51, 62, 74, 117]. The general structure of the proposed algorithm, that we call GRASP TNDP, is the following (implementation details are given in Section 5.4):

- The construction procedure uses the Pair Insertion Algorithm (Chapter 4). It generates a set of routes R, which satisfies constraints (5.3) and (5.4) of the optimization problem $\mathcal{P}$. Routes are constructed by using shortest paths between vertices in G and then inserting additional pairs of vertices into them.

- The local search calculates a near optimal set of frequencies F, according to constraints (5.5) and (5.6), for a given trade-off level between the conflicting objectives represented by functions (5.1) and (5.2). This procedure takes a random vector of weights and uses a neighborhood structure to advance in the direction of improvement of a single composite objective function. The neighborhood of a solution is defined by varying its frequencies in a predetermined set.

In this way, at each GRASP iteration, different points in both decision and objective spaces are sampled. Different trade-off levels are obtained by varying from one GRASP iteration to another, parameters of maximum route duration t_{max} at the construction procedure and a random vector of weights λ at the local search (Section 5.4). All solutions of the trajectory of the local search are added to the set of potentially non-dominated solutions under construction P. At the end of each GRASP iteration, all dominated solutions in P are deleted. The assignment model of Baaj and Mahmassani [8] is used to load the demand onto a given solution S, to evaluate objective function $Z_1(S)$ and to verify frequency feasibility according to constraint (5.6).

5.4 The algorithm

When we instantiate the GRASP metaheuristic for a particular application, we have to tailor all its problem dependent aspects. The construction algorithm has to be specified, which entails to specify how to build the list of candidate elements to be added to the solution under construction, how these elements are ranked at each step of the construction according to a required adaptive greedy function, how to construct the restricted candidate list, and how the elements are selected from that list. For the local search, a neighborhood structure and its exploration strategy have to be defined. Also a stopping rule is required. We now discuss all these elements for our solution to the TNDP.

5.4.1 Construction algorithm

We use the greedy randomized variant of the Pair Insertion Algorithm (PIA), which produces a set of routes for the TNDP. Here we present its relevant aspects, more details can be found in Chapter 4. The construction algorithm (Figure 5.2) starts with an empty set of routes R, and iteratively seeks to satisfy the demand specified by the origin-destination matrix D. At each iteration step, a restricted candidate list rcl is constructed by selecting the $\alpha|l|$ pairs of vertices (i, j) with highest demand d_{ij} in l, where $\alpha \in [0..1]$ is a real-valued parameter of GRASP and l is a list made of all pairs of vertices whose demand is

```
procedure Construction(in D_0^min, D_01^min, in ρ_max, t_max, in α, out R);
  R ← ∅; D_0(S) ← 0; D_01(S) ← 0;
  l ← List of pairs of vertices (i, j) of G with d_ij ≠ 0;
  while D_0(S) < D_0^min or D_01(S) < D_01^min do
     rcl ← Construct according to α and l;
     (u, v) ← Select randomly from rcl;
     r ← Create a route with the shortest path between u and v in G;
     r' ← Create a route by inserting u and v in the most convenient
           positions in the most convenient route r'' in R;
     if cost(r) < cost(r') − cost(r'') then
        R ← R ∪ {r};
        Delete from l pairs of vertices whose demand is covered directly by r;
     else
        R ← R ∪ {r'} − {r''};
        Delete from l pairs of vertices whose demand is covered directly by r';
     end if;
     Update D_0(S) and D_01(S);
  end while;
  Filter routes in R;
  return R;
end Construction;
```

Figure 5.2: Construction algorithm

not yet satisfied (directly) by routes in R. The pair of vertices (u, v) is randomly selected from rcl, and its corresponding demand d_{uv} is satisfied according to one of the two cases explained in Section 4.4. When inserting pairs of vertices on existing routes, constraints of maximum duration t_{max} and maximum circuity factor ρ_{max} are applied, as explained in Section 4.4. The construction ends when constraints (5.3) and (5.4) are satisfied. The algorithm finally performs an operation that filters routes that are completely included in other ones.

5.4.2 Local search

The local search operates with the set of frequencies $F = \{f_1, \ldots, f_{|R|}\}$ of a solution $S = (R, F)$; this means that only frequencies are decision variables in this phase (Figure 5.3). The domain of frequencies is discretized in the set $\Theta = \{\theta_1, \ldots, \theta_{|\Theta|}\} \in \mathbb{R}^{|\Theta|}$ as it is done in Section 3.2.3. That set is a given parameter that is sorted in increasing order, satisfying $\theta_1 \geq f_{min}$ and $\theta_{|\Theta|} \leq f_{max}$. The neighborhood N_S of S is obtained by varying the frequency of every route in S. A solution S' is a neighbor of S if both solutions have exactly the same routes, they differ in the frequencies of one route and these frequencies are consecutive values in Θ.

According to this neighborhood definition, the local search algorithm evaluates the costs of increasing or decreasing the frequencies in all routes of solution S. At each step of the local search, the cardinality of N_S can be at most $2|R|$. However, this number can be smaller when there are routes with frequencies equal to θ_1 or $\theta_{|\Theta|}$. The local search receives a random vector of weights $\lambda = (\lambda_1, \lambda_2)$ and successively moves forward

```
procedure LocalSearch(in λ, in S, in out P);
  current ← S;
  P ← P ∪ {current};
  stop ← false;
  repeat
     S′ ← FirstImprovement(current, λ);
     if S′ better than current then
        current ← S′;
        P ← P ∪ {current};
     else
        stop ← true;
     end if;
  until stop;
  return P;
end LocalSearch;
```

Figure 5.3: Local search

to the neighbor which minimizes the composite objective function $\lambda_1 Z_1 + \lambda_2 Z_2$, using a first improving strategy [104]. This means that whenever a neighbor that improves the objective value of the current solution is found, the exploration of the neighborhood is terminated and the local search proceeds towards its next step. The evaluation of each neighbor solution involves an invocation to the algorithm that implements the assignment model (Section 5.4.4). Solutions that violate constraint (5.6) are discarded.

5.4.3 GRASP TNDP

Figure 5.4 presents a pseudo code of the GRASP TNDP algorithm. It begins by calculating the shortest path between all pairs of vertices in G. This is done just once, independently of the GRASP iterations, because the cost of the edges of G are considered as constant, not depending on the flows produced by different solutions. The maximum duration of routes t_{max} is determined at each GRASP iteration by sampling a random uniform value in the real interval $[t_{max}^{ini}, t_{max}^{end}]$ (given parameters). This idea is applied to obtain diverse solutions, each having internally homogeneous characteristics, all routes having approximately the same duration. The initial frequency of every route in R is set as the maximum value of the set Θ; thus we try to find an initial solution that is feasible with respect to constraint (5.6). The random vector of weights $\lambda = (\lambda_1, \lambda_2)$ is determined by sampling a random uniform value in the real interval $[0, 1]$ for λ_1 and then setting $\lambda_2 = 1 - \lambda_1$.

Observe that GRASP TNDP is consistent with the underlying bilevel structure of formulation (5.1)-(5.6). In particular, given a solution, its objective value according to (5.1) and the fulfilment of constraint (5.6) are known after applying the assignment submodel. We should mention that demand covering constraints (5.3) and (5.4) have a slightly different meaning than (3.60) in formulation OPT3: while in GRASP TNDP they are intended to be ensured by the construction algorithm PIA, they may be violated after applying the assignment submodel (Section 5.4.4). This may happen because PIA uses a preliminary

```
procedure GRASP_TNDP(in D_0^min, D_01^min, in ρ_max, t_max^ini, t_max^end,
                     in NumIterations, α, out P);
  Calculate shortest paths between all pairs of vertices in G;
  P ← ∅;
  for i = 1 to NumIterations do
     t_max ← Random uniform value in [t_max^ini, t_max^end];
     Construction(D_0^min, D_01^min, ρ_max, t_max, α, R);
     F ← Initial frequencies;
     S ← (R, F);
     λ ← Random vector of weights;
     LocalSearch(λ, S, P);
     Delete dominated solutions of P;
  end for;
  return P;
end GRASP_TNDP;
```

Figure 5.4: GRASP TNDP algorithm

assignment for checking demand covering constraints; that assignment has less information about the solution (it does not know the values of frequencies).

5.4.4 Assignment submodel

The assignment model of Baaj and Mahmassani [8] is used to load the demand D among the lines of a given solution S; it is needed in order to calculate $Z_1(S)$ and to verify constraint (5.6). In general terms, it can be considered as a variant of the assignment model of optimal strategies [110] (explained in Section 3.2.1). The main differences with respect to that model are:

- It adopts the criterion proposed in [60] concerning transfers. The users consider a lexicographic strategy in the choice among competing routes (different lines connecting the same OD pair), transfer minimization being the primary choice criterion. Observe that in optimal strategies, transfers are ignored; they are implicitly codified in the trajectory graph.
- A heuristic solution of the problem of travel time minimization is performed, instead of an exact one.

The model performs an explicit enumeration of the different routes connecting every OD pair, including transfers if necessary. It does not consider congestion effects neither in the calculation of on-board travel time tv (which is calculated in terms of fixed cost c_e on edges e) nor the waiting time tw (solutions lacking bus capacity are discarded). Calculations are performed as follows, for OD pair (i, j). Let R_{ij} be the set of routes connecting i and j; assume that it is not empty, therefore transfers are not needed. Let p_{ij}^k be the proportion of the demand d_{ij} assigned to route $r_k \in R_{ij}$ (i.e., the frequency share rule (2.2)), defined as

$$p_{ij}^k = \frac{f_k}{\sum_{r_m \in R_{ij}} f_m} ,$$

which is used to compute the following values:

- $tv_{ij} \leftarrow \sum_{r_k \in R_{ij}} p_{ij}^k t_{ij}^k$, where t_{ij}^k is the on-board travel time from i to j using r_k,
- $v_a \leftarrow v_a + p_{ij}^k d_{ij}$ for each arc a of each route $r_k \in R_{ij}$ that is used to transport the demand d_{ij}.

The waiting time is computed as $tw_{ij} \leftarrow 1/(2\sum_{r_k \in R_{ij}} f_k)$, which corresponds to expression (2.1) with $\beta = 1/2$,

If set R_{ij} is empty, transfers are considered; corresponding calculations are similar to the case without transfers. Transfer time tt_{ij} is set to σ_t in order to penalize each transfer performed by demand d_{ij}.

5.5 Numerical results

We test the GRASP TNDP algorithm with the following scope:

1. Investigate whether the proposed algorithm produces a set of diverse non-dominated solutions for the TNDP (Section 5.5.1). Some descriptive measures are calculated in order to illustrate the diversity of the obtained results. Also these measures may be useful to the decision maker.
2. Compare the results of GRASP TNDP with other results published in the literature (Section 5.5.2).
3. Compare the relative efficiency of the multi-objective algorithm with respect to a single-objective variant, used as subroutine in the Weighted Sum Method (Section 5.5.3).
4. Apply the GRASP TNDP algorithm to a real city, comparing its solutions with the solution of the public transportation operating in the city (Section 5.5.4).

We use two test cases to perform the experiments:

- Mandl, taken from [9]. Its corresponding graph has 15 vertices and 21 edges representing a real city. Its origin-destination matrix is very dense, having 76% of non-zero elements. The case proposed by Mandl has been used as benchmark instance by several authors who studied the TNDP [9, 103, 123].
- Rivera, constructed in the context of a project related to this thesis (Appendix A). It corresponds to a small city of 65,000 inhabitants in Uruguay. Its graph has 84 vertices and 143 edges and its origin-destination matrix was obtained from real data, having 5% of non-zero elements.

We note the difficulty of performing a comprehensive experimental study over a set of many different test cases of the TNDP. Both the model and the algorithm proposed on this work have several parameters which must be set or adjusted in order to perform the experiments in a realistic and coherent scenario.

Parameter	Mandl	Rivera	Units
D_0^{min}	0.5	1.0	-
D_{01}^{min}	1.0	1.0	-
t_{max}^{ini}	40	40	minutes
t_{max}^{end}	120	120	minutes
σ_t	5	5	minutes
ω	40	28	seats
η	1.25	1.50	-
ρ_{max}	1.5	1.5	-
f_{min}	1/60	1/60	vehicles/minute
f_{max}	1/2	1/2	vehicles/minute
α	0.2	0.2	-

Table 5.1: Parameter configuration

The used parameter configuration is shown in Table 5.1; the values were set as follows. D_{01}^{min} is set in order to satisfy the whole demand. For Mandl we set $D_0^{min} = 0.5$ in order to compare with results published in the literature; for Rivera we set $D_0^{min} = 1.0$, since in the city the whole demand is satisfied without transfers (note that it is a strong requirement on demand satisfaction). Both extremes of t_{max} were set with reasonable values for the dimensions of both cases and their values of travel time (represented by the costs of the edges of G). Parameters σ_t, ω and η are taken from [9] for Mandl, while for Rivera we use the values of ω and η observed from the real system. The value of ρ_{max} is suggested in [10]. We use realistic values for frequency range and set of frequency values. The domain of frequencies was discretized in the set $\Theta = \{1/60, 1/50, 1/40, 1/30, 1/20, 1/10, 1/5, 1/2\}$. The α parameter of the randomized version of PIA was previously adjusted. The algorithm was coded in C++ and all tests were carried out on a Pentium 4 PC, with a 2 GHz processor and 2 GB of RAM.

5.5.1 Results of GRASP TNDP

The execution of the GRASP TNDP algorithm for the case of Mandl with parameter configuration given in Table 5.1 and 1000 GRASP iterations took 245 seconds and produced an approximated Pareto front composed of 96 non-dominated solutions. Table 5.2 shows for 10 solutions (selected as representative points from different regions of the front), the corresponding values of Z_1 (along with its components tv, tw and tt), Z_2, number of routes $|R|$ and averaged values over each route of headway $1/f$ and duration t; values of the last two columns are expressed in minutes.

From Table 5.2 we can observe that a wide range of trade-off levels between the conflicting objectives is covered, specially when looking at column Z_2 (fleet size). The trade-off can be characterized by the values of the number of routes and the average of the route headways. In this way, although there is no monotonic tendency, we can say that solutions with low cost for the users (and therefore with high cost for the operators) are characterized by high values of $|R|$ and low values of $1/f$, and vice versa. Moreover, we can observe that variation along the Pareto front is higher in the waiting time component than in the

Solution	Z_1	tv	tw	tt	Z_2	$\|R\|$	$1/f$	t
1	122.96	110.76	7.44	4.76	189.00	10	2	38
2	133.01	114.84	13.23	4.93	84.00	6	4	44
3	138.55	113.64	19.98	4.93	61.80	6	6	44
4	147.44	117.87	24.89	4.69	38.80	4	5	49
5	159.88	119.47	35.65	4.76	27.00	3	8	67
6	172.33	116.66	52.26	3.40	19.30	4	15	63
7	185.68	121.54	58.28	5.87	14.40	3	13	59
8	221.43	122.87	94.25	4.31	9.20	3	20	61
9	279.99	121.01	149.40	9.58	6.10	5	32	35
10	381.37	119.07	253.24	9.06	4.08	6	52	34

Table 5.2: Results of GRASP TNDP

on-board travel time component of Z_1.

We also present several measures that are used to illustrate how diverse the results produced by the algorithm are. Two of them are relative to distances to lower bounds from the users viewpoint (equations (5.7) and (5.8)) and the other two are relative to the utilization of buses. All these measures may be used by the decision maker as a guide in the selection of one non-dominated solution or in the evaluation of an existing one.

From [9] we take the idea of an optimal route set from the users viewpoint, which allows every pair of vertices (i, j) to transport its demand d_{ij} along the shortest path in G (independent of any route), with cost t^*_{ij}. According to this, a lower bound for on-board travel time is defined as

$$tv^* = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} d_{ij} t^*_{ij} \ . \tag{5.7}$$

We propose an analog definition of lower bound for the waiting time. An optimal frequency set from the users viewpoint can be defined when every pair of vertices is served by a route with frequency equals to f_{max}. This is actually a pseudo lower bound, since there can exist solutions where some pairs of vertices are served by more than one route with the maximum frequency (see Section 5.4.4 for an explanation on waiting time calculation).

$$tw^* = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \frac{d_{ij}}{2 f_{max}} \tag{5.8}$$

Table 5.3 shows for the same solutions of Table 5.2, values of distances from tv and tw to tv^* and tw^*, respectively, where the distance from a value v to its lower bound v^* is defined as $dist(v, v^*) = (v - v^*)/v^*$. Table 5.3 also shows measures relative to the utilization of buses, averaged over each route, namely:

- Mean utilization $\bar{U}$, defined for each route r_k as $\bar{\phi}_k/(f_k \omega)$, where

$$\bar{\phi}_k = \frac{\sum_{a \in r_k} v_a c_a}{\sum_{a \in r_k} c_a} \ .$$

Solution	$dist(tv, tv^*)$	$dist(tw, tw^*)$	$\bar{U}$	U^*
1	0.02	-0.31	0.01	0.02
2	0.06	0.22	0.03	0.04
3	0.05	0.85	0.04	0.06
4	0.09	1.30	0.07	0.09
5	0.10	2.30	0.11	0.17
6	0.08	3.83	0.13	0.19
7	0.12	4.39	0.19	0.26
8	0.14	7.72	0.34	0.44
9	0.12	12.82	0.47	0.62
10	0.10	22.43	0.70	0.92

Table 5.3: Results of GRASP TNDP, additional measures

- Critical utilization U^*, defined for each route r_k as $\phi_k^*/(f_k\omega)$.

In Table 5.3 we can observe that the trade-off level between the conflicting objectives also can be characterized by values of distances to lower bounds and utilization of buses. In this way, solutions with low cost for the users are characterized by low values of all these four measures, and vice versa.

The distance to the lower bound in the on-board travel time component is no greater than 0.14. Nevertheless, the distance to the lower bound in the waiting time component is up to 22.43. We can observe that the solution with lowest cost for the users present a negative value in $dist(tw, tw^*)$, since tw^* is actually a pseudo lower bound; that solution has a high average of frequencies (see column $1/f$ in Table 5.2). It is worth mentioning that this is rather a theoretical result, since from a practical viewpoint it may not be possible to operate several lines with high frequencies on the same edge of the network; in other words, the constraint of street capacity is not included in the formulation.

Moreover, we can observe that values of both mean and critical utilization of buses are lower than 1.00 for all solutions. Maximum critical utilization is shown by solution 10; this value can not be greater than the level specified by the maximum load factor parameter η, 1.25 for the case of Mandl.

5.5.2 Comparison with results published in the literature

In this section we compare the results of GRASP TNDP with results published by Baaj and Mahmassani [9] using the benchmark test case of Mandl. Although other authors [123] have used the same case the results are not comparable because they solve a different optimization problem. Note that since the comparison is done in terms of objective values, also the same assignment model should be used.

The results of Baaj and Mahmassani [9] were obtained using a greedy algorithm (Route Generation Algorithm) followed by a local improvement phase (Route Improvement Algorithm); the solutions were evaluated using the Transit Route Analyst algorithm [8], which implements the assignment model described in Section 5.4.4. These algorithms solve the same optimization problem considered in this work (except for the upper limit on fre-

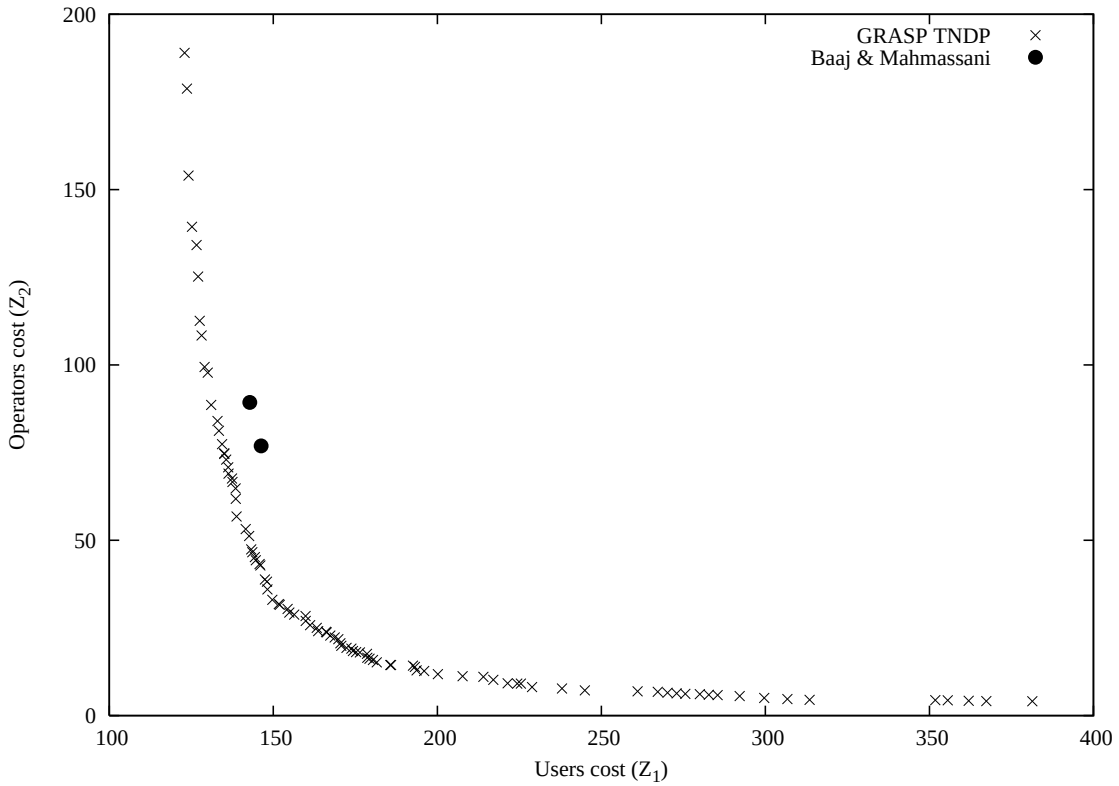


Figure 5.5: Comparison with results of Baaj and Mahmassani

quency range f_{max} which is not considered in [9]); the difference is the adopted approach to handle the multi-objective aspect of the problem. We use the same parameter configuration except for f_{min} which is not specified in [9]. We note that the algorithms proposed in that reference work do not perform an explicit search in the domain of frequencies, which are set as the minimum value that satisfies constraint (5.6) of maximum load factor for each route.

Figure 5.5 shows the Pareto front obtained by GRASP TNDP (Section 5.5.1) as well as the two solutions (BM) obtained by Baaj and Mahmassani [9] for $D_0^{min} = 0.5$. The values of Z_1 published in the reference work were scaled in order to use the same units, since they are calculated directly from the origin-destination matrix expressed in trips per day, while we use the values expressed in trips per minute. We can observe that solutions BM are dominated by solutions of GRASP TNDP. Moreover, solutions BM are concentrated on a particular level of trade-off between the conflicting objectives. It is worth mentioning that without additional information we can not state which part of the Pareto front corresponds to practicable solutions.

5.5.3 Comparison with the Weighted Sum Method

The aim of this experiment is to study the computational efficiency of the proposed multi-objective approach to the TNDP. We compare the performance of the multi-objective

algorithm with a single-objective version of it. We use the Weighted Sum Method [31] as reference for comparison since it is a straightforward way to obtain a set of non-dominated solutions using an approach that formulates a weighted sum of objectives (most existing models in the literature for the TNDP). When applied to problem $\mathcal{P}$ (Section 5.2), the Weighted Sum Method consists in minimizing the objective function (5.9) under constraints (5.3)-(5.6), for a given set of different vectors of weights $\lambda = (\lambda_1, \lambda_2)$,

$$Z(S) = \lambda_1 Z_1(S) + \lambda_2 Z_2(S) \; . \tag{5.9}$$

We implemented a single-objective version of the GRASP TNDP algorithm presented in Section 5.4, to be used as subroutine in the classical Weighted Sum Method. The single-objective algorithm differs from its multi-objective counterpart in the following aspects:

- It produces a single solution at every run.
- It receives a vector of weights $\lambda = (\lambda_1, \lambda_2)$, where $\lambda_1 + \lambda_2 = 1$, representing the relative importance between the conflicting objectives Z_1 and Z_2, which is used for setting $t_{max} = t_{max}^{ini} + \lambda_2(t_{max}^{end} - t_{max}^{ini})$ in the construction procedure and composing a single objective function $Z = \lambda_1 Z_1 + \lambda_2 Z_2$ in the local search (for all GRASP iterations).

In order to compare both algorithms we proceed as follows. The single-objective algorithm is run for m different vectors of weights (evenly distributed in the interval $[0, 1]$), each run having the same number of GRASP iterations ($NumIterations$). Then, for the (multi-objective) GRASP TNDP we assign a number of GRASP iterations equal to $m \times NumIterations$; thus, both algorithms execute the same number of construction and local search phases, representing somehow the same computational effort. We run both algorithms for different combinations of m and number of GRASP iterations. For each combination we calculate the non-dominated set of the result of merging the Pareto fronts produced by both algorithms. The number of surviving solutions from this process is presented in Table 5.4, where each entry shows for the cases of Mandl and Rivera: (a) number of solutions coming from the Weighted Sum Method (WS), (b) number of solutions coming from the multi-objective approach (MO) and (c) the ratio b/a, as a measure of relative efficiency (re) of MO. For each test case, the last row of its corresponding part of the table shows the overall relative efficiency of MO calculated as the average of all re over the different values of m for a given value of GRASP iterations.

The main conclusion from these results is that MO produces more non-dominated solutions than WS, with the same computational effort. Note that without additional information it is not possible to determine which solutions are better than others, since all of them are non-dominated. Moreover, we can observe that the overall relative efficiency of MO seems to increase as we increase the number of GRASP iterations used in the single-objective algorithm. This means that whenever we decide to increase the accuracy of the approximated results, it is more beneficial to adopt the multi-objective approach. Figure 5.6 shows the Pareto fronts obtained by both algorithms for Rivera ($m = 10$ and 500 GRASP iterations) and the solution corresponding to the real public transportation system of the city (that we call as *reference solution*); note that this solution can be used to identify the level of trade-off where practicable solutions are located.

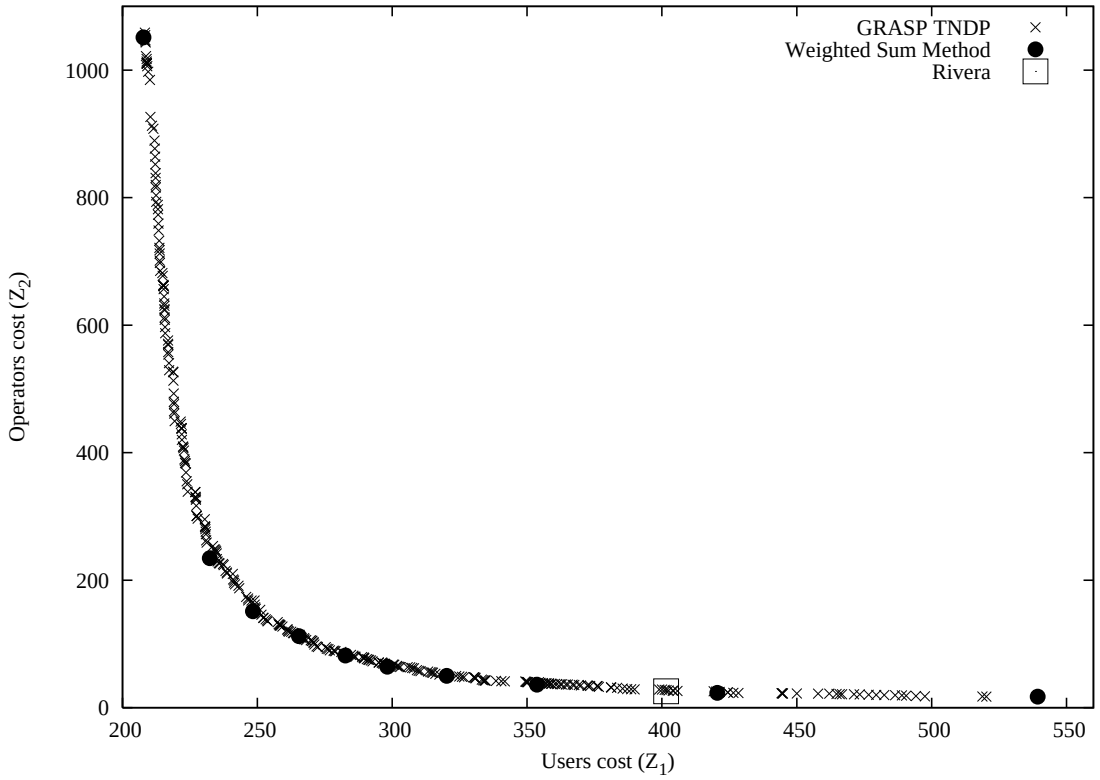


Figure 5.6: Non-dominated solutions obtained by both algorithms for the case of Rivera

	Number of GRASP iterations														
m	10			50			100			200			500		
Mandl															
2	1	34	34.00	2	46	23.00	2	46	23.00	2	67	33.50	1	88	88.00
5	3	35	11.67	5	50	10.00	5	68	13.60	4	86	21.50	3	118	39.33
10	3	43	14.33	7	69	9.86	7	83	11.86	8	95	11.88	8	124	15.50
avg. re			20.00			14.29			16.15			22.29			47.61
Rivera															
2	2	174	87.00	2	228	114.00	2	242	121.00	2	265	132.50	2	288	144.00
5	5	190	38.00	5	235	47.00	5	252	50.40	5	272	54.40	5	276	55.20
10	9	199	22.11	10	233	23.30	10	242	24.20	9	241	26.78	10	277	27.70
avg. re			49.04			61.43			65.20			71.23			75.63

Table 5.4: Results of Weighted Sum Method and multi-objective approach

The tendency of the overall relative efficiency is monotonic for the case of Rivera; for Mandl, the observation is valid for high values of GRASP iterations. We used small values of m since we have limited computational resources (one GRASP iteration takes 0.25 seconds for Mandl, while it takes 25 seconds for Rivera). For real cases, although we are not assuming anything concerning the method used by the decision maker to analyze the set of non-dominated solutions, we consider that it is not desirable to use a high number of different vectors of weights in the Weighted Sum Method.

In order to investigate the behavior of the algorithms for a high number of weight vectors, we tested the case $m = 100$ for Mandl, and we observed the following progression (30.67; 9.62; 7.95; 9.75; 9.82) of the relative efficiency according to the number of GRASP iterations (10; 50; 100; 200; 500). Comparing with the rows corresponding to low values of m, this suggests that whenever we allow the Weighted Sum Method a denser coverage of the objective space (high values of m), the multi-objective approach will need more computational effort (GRASP iterations) to improve its relative efficiency.

5.5.4 Application to a real case

In this experiment we use the case relative to the city of Rivera (Appendix A), to compare the results produced by the GRASP TNDP algorithm against the reference solution, i.e., the solution operated by the public transportation system of the city. This solution was codified in terms of the same graph G used to execute GRASP TNDP and was evaluated using the same assignment model (Section 5.4.4) and parameters. We note that the theoretical value corresponding to Z_2 (fleet size) of the reference solution is 25.65 (Table 5.5), while its real value is 23. Despite the fact that the theoretical value (equation (5.2)) is a fractional approximation of a number that in reality is integer (number of buses), the discrepancy is due to errors incurred in the modeling of the costs (travel times) of the edges of G, which are calculated in terms of a constant average bus speed, while in Rivera the bus speed varies among different lines.

Table 5.5 shows for 20 solutions that represent different regions of the approximated Pareto front produced by GRASP TNDP (that we call P), values of Z_1, Z_2, number of routes $|R|$ and route headway $1/f$ and duration t (both in minutes) averaged over the routes of each solution. We can observe that the algorithm produced solutions in a wide range of trade-off levels. In the extreme corresponding to solutions of low cost for the users, we obtained a minimum value $Z_1 = 208.32$, which is very close to the sum of the lower bounds of on-board and waiting time (see Section 5.5.1), in this case 210.64; note that tw^* is a pseudo lower bound, since it can be improved by solutions having many routes with high frequencies, as it is the case of route number 1 in Table 5.5. In the opposite extreme, we obtained a value $Z_2 = 17.24$ with 13 routes, which is significantly less (considering the dimensions of the case) than its corresponding value in the reference solution ($Z_2 = 25.65$).

Moreover, GRASP TNDP produced solutions which are very close to the reference solution in the objective space. The solution of P that is closest to the reference one is 0.7% and 3.1% worse than such solution, in terms of Z_1 and Z_2 respectively; Figure 5.7 shows a zoom of Figure 5.6 in that region of the objective space. This fact should be analyzed taking into account the approximate nature of the algorithm and the source of

the data used to construct the test case. Since we use an approximate algorithm, the front P could be improved in the sense of the closeness to the optimal Pareto front; we can assume that there are solutions that dominate some solutions of P, which could not be found by the algorithm. If we assume that GRASP TNDP found solutions that are very close to the optimal Pareto front, the closeness of the reference solution could be explained by two reasons:

1. The reference solution is a good one, since it has been continuously adapted during the existence of the public transportation system, by planners who have a deep local knowledge of the reality.

2. The origin-destination matrix used to solve the optimization model was estimated from a survey done on-board the buses of the lines of the reference solution (Appendix A). Despite the fact that this matrix is considered as a good approximation to the matrix of desired trips, the demand estimated by this method is somehow strongly adapted to the supply represented by the lines of the reference solution. Then, we can expect that the evaluation of that solution in terms of objective functions (5.1) and (5.2) is good.

Solution	Z_1	Z_2	$\|R\|$	$1/f$	t
1	208.32	1058.83	49	2	43
2	210.35	926.48	39	2	48
3	213.35	748.52	50	7	44
4	215.47	634.02	50	5	43
5	218.87	512.99	49	7	44
6	222.60	406.95	38	7	49
7	227.51	300.83	46	9	44
8	234.77	244.92	41	12	47
9	243.27	187.28	47	16	44
10	257.89	130.87	40	18	48
11	267.51	107.08	24	18	61
12	281.10	87.43	19	18	70
13	295.11	70.69	18	25	76
14	310.49	57.38	19	28	62
15	334.19	42.69	19	36	70
16	358.19	37.73	15	34	73
17	376.14	33.75	17	45	71
18	419.21	25.67	12	44	87
19	466.81	21.01	13	55	86
20	520.27	17.24	13	60	80
Reference	401.56	25.65	13	37	63

Table 5.5: GRASP TNDP applied to Rivera

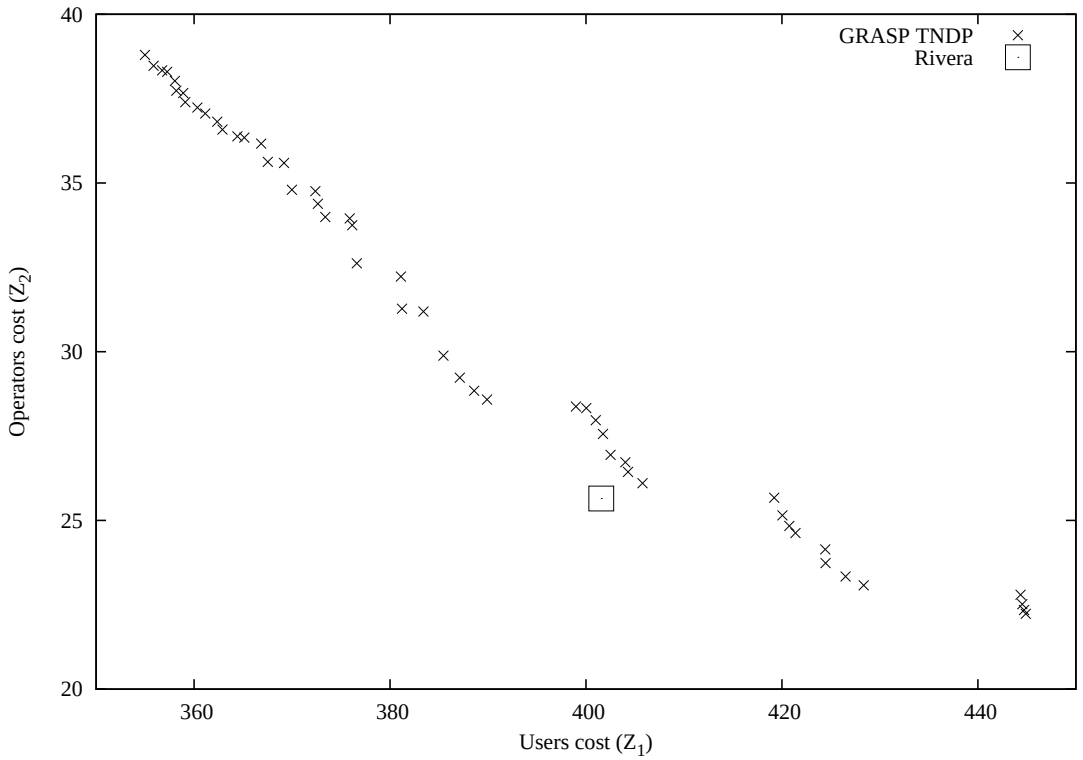


Figure 5.7: Results of GRASP TNDP around the reference solution

5.6 Conclusions and future work

We have proposed an algorithm based on the GRASP metaheuristic to solve the TNDP with a multi-objective approach. We show that the proposed multi-objective algorithm produces a diverse set of non-dominated solutions in a single run. Existing and proposed measures are presented, which can be useful for the decision maker to characterize a given solution of the Pareto front, with respect to the interests of both users and operators. The solutions obtained dominated other solutions published in the literature for the benchmark case of Mandl; for the real case of Rivera, the algorithm produced solutions comparable with the solution operated by the public transportation system of the city. We also show that the multi-objective approach is more efficient than the Weighted Sum Method, in the sense that it produces more non-dominated solutions with the same computational effort.

For future research, we identify several directions. The hypothesis of inelastic demand is used to simplify the model. However, for some cases elastic demand must be considered in order to model the changes in the origin-destination matrix according to the supply of public transport, specially for solutions in the extremes of the Pareto front. Elastic demand has been incorporated to the TNDP in [41, 61, 72] and a challenging work consists in trying to incorporate this characteristic into the presented multi-objective approach.

We note that the presented numerical results lack of an evaluation of the closeness to the optimal Pareto front (which was not available). A possible way to accomplish that

evaluation consists in implementing a modified version of the GRASP based algorithm to solve a different optimization model (for example the problem stated by formulation OPT2 presented in Section 3.2.3).

Most metaheuristics for the TNDP are implementations of Genetic Algorithms [94, 103, 113] for single-objective optimization. These ideas can be used to design multi-objective versions, since there are several applications of genetic algorithms for multi-objective optimization [24, 31]. This line of research was explored in [5] in the context of this thesis.

We observe that parallel implementations have improved the performance of several metaheuristic based algorithms; this is the case of many GRASP based algorithms [104] as well as the parallel Genetic Algorithm for the TNDP proposed in [1]. A parallel version of the GRASP TNDP algorithm could improve the performance of its original version; this could be useful when applying the algorithm to instances of realistic size.

Chapter 6

Final discussions and conclusions

We have studied models and algorithms for the optimal design of bus routes in urban public transportation systems. Our contributions can be expressed in a very condensed way as: (i) a mathematical programming formulation that includes several realistic characteristics of the problem, which can be used in the future to propose an exact solution approach and (ii) a heuristic approach that can be potentially applied to real cases related to small to medium-sized cities, which admits some improvements in order to obtain better results. The contributions related to (i) are supported by directly observing the elements of the problem included in our formulations, in relation to the state of the art. On the other hand, the contributions related to (ii) are supported by numerical results which are compared with results coming from the state of the art as well as from real solutions.

In this chapter we formulate conclusions and we identify future work about the overall research work concerning this thesis; previous chapters have their own specific conclusions. We conclude about the different proposed methodologies and our experience concerning the numerical tests and the application to a real case. Finally we give some opinions related to the application of the different methodologies to the TNDP and the evolution of this research field. We also give some recommendations related to the application of the proposed methodologies to real cases.

6.1 The methodologies

The main contributions of this thesis concerning the methodologies are an explicit mathematical programming formulation to model the problem, a greedy constructive algorithm to obtain part of a solution to the problem and a metaheuristic that solves approximately the problem modeled with a multi-objective approach. An effort was done in order to position the problem within the field of mixed integer linear programming (MILP). Thus we were able to apply theoretical properties as well as efficient solution methods, since MILP is an extensively studied and developed area. Furthermore, when adding additional constraints to the problem, we used the framework of bilevel programming as a rich tool that enables to model naturally many characteristics of the problem (mainly the interactions between the planning entity and the users of the system) and helps to devise solution techniques. Concerning the developed heuristics, our strategy was first to design a constructive algorithm which can quickly produce solutions of reasonable objective

values, handling complex constraints like those related to transfers. Many computational experiments were carried out in order to test the ability of the algorithm to generate good solutions. The multi-objective nature of the problem inspired an approximate solution method that produces solutions in a wide range of trade-offs between the interests of users and operators. The proposed method exploits the multi-objective nature of the TNDP in order to obtain efficiently all those solutions. The constructive algorithm is used as subroutine of this method.

Mathematical formulation

The main contribution concerning the mathematical modeling is the inclusion of an assignment model that represents in a realistic manner the behavior of the users in systems based on buses. We use the optimal strategies assignment model [110] which fulfills this requirement. Thus, the main difference of the proposed model of route optimization with respect to the existing ones is the consideration of the waiting time and multiple lines, two aspects of the problem that are closely related [33]. We also present a discussion about the inclusion of important constraints and its impact in the mathematical structure of the problem. The proposed formulations were used to (i) obtain optimal solutions for very small instances of the problem, (ii) validate results of heuristics, (iii) reason about the structure of the problem and the impact of adding new constraints and (iv) apply the model to a decision making situation in the context of a real case related to a small city. We identify three different directions for future work: (i) development of an exact approach based on the formulation, (ii) development of algorithms to solve the problem in the cases where the formulation has a bilevel structure, and (iii) dealing with the size of the model when adding transfer constraints. We also note that the proposed formulation allows to work with different levels of aggregation; thus, it can be used to model the public transportation system either only in terms of centroids and connections among them, as well as in terms of the detailed street network and access/egress arcs. At the most detailed level, decisions concerning location of bus stops and design of limited-stop routes could be incorporated; these are possible applications, other than the specific subject of this thesis. Moreover, related problems like the frequency optimization can benefit from our proposed formulation.

Constructive algorithm

We proposed a greedy constructive algorithm called PIA, to obtain a set of routes that takes into account the interests of users and operators and demand covering constraints. The results obtained improve the ones existing in the literature. The algorithm has desirable properties, namely: (i) it produces solutions comparable with real ones, (ii) it does not require the application of a complex assignment model and (iii) its logic is simple and may be understood by the planner. We consider (i) as an important issue concerning the validation of the algorithm, which was tested with a real case using real data. Concerning (ii), we have proposed an algorithm that can be used as subroutine to generate quickly an initial solution; further improvement methods (for example metaheuristics) can be applied, as the one proposed in [4]. Finally, (iii) contributes to the use of PIA as a systematic procedure that can be used even interactively by the planner in order to design bus routes.

Multi-objective metaheuristic approach

Although the main objective when designing routes for public transportation systems is the maximization of the level of service offered to the users [33], other interests should be considered in order to design a sustainable system. The modeling of this aspect of the problem using concepts of multi-objective optimization has been treated scarcely in the literature. Moreover the efficiency of solution methods to solve the TNDP as a multi-objective problem has not been discussed. The problem is hard to solve even if we consider a single objective, therefore approximate methods (heuristics) should be used for cases of realistic size. When a multi-objective approach to the problem is adopted, we have to design efficient multi-objective heuristics to solve it. In this thesis we propose a metaheuristic called GRASP TNDP to solve the problem of route optimization with a multi-objective approach. GRASP TNDP relies on the route construction algorithm PIA. Using the benchmark case of Mandl, we showed that the algorithm improves the results published in the literature. Using a real test case related to the city of Rivera, we also compare results of GRASP TNDP with the solution that is operated by the public transportation system of the city. Our closest solution to the real one is worse than a such solution by a small percentage; however, that percentage (3.1% in the operator objective function) can be significant when it is accumulated in a long term period. Concerning this comparison we should take into account that: (i) the solution of Rivera contains circular lines, therefore it is out of the decision space bounded by the hypothesis that rules our algorithms (Section 2.1.3); (ii) our solution admits to be improved since its route structure relies only in a greedy construction. Thus, the algorithm proposed in [4] could be used to improve the solution by changing its route structure.

General comments

In the following we elaborate general conclusions about the different methodologies to tackle the TNDP proposed in this thesis, in relation to the state of the art.

Some existing exact approaches to the problem are restricted to applications to cases lacking real characteristics. The models proposed in [58, 120] are applied to very small test instances of unrealistic size. These works can be considered as contributions to the modeling of the problem. In [107], a real test case related to the long distance network of the German railway is used. However, only solutions to the linear relaxation of the model are found (spending a running time of two and a half hours); no method is proposed to find an integer solution. Although no practical method is proposed on that work, it can be considered as a step in the development of exact methods to the TNDP. In [15] an approximate solution is obtained for Postdam, a city which had 27 bus lines and 4 tram lines when the study was undertaken. The solution method first solves the linear relaxation using a decomposition technique and then obtains an integer solution by using a heuristic procedure. In their experiments, the authors obtain a fractional optimal solution in less than 10 minutes on a Pentium 4 machine of 3.4 GHz processor. The obtained (approximate) integer solution has a gap with respect to the fractional one of around 50%; this gap depends on the values used to weight each term of the objective function. Our contribution with respect to those studies is the modeling of aspects of the problem that are relevant to systems based on buses: the waiting time and the assignment to multiple

lines. Using our formulation, optimal integer solutions can be found for cases of sizes comparable to those used in [58, 120]. Using the case of the city of Rivera (whose public transportation system has 13 lines), we were not able to compute an integer solution along with a lower bound as it is done in [15]; we do not have information beforehand to decide which lines are likely to be part of a good solution. In our experiments we use the formulation with a particular purpose, other than obtaining the optimal solution: we generate a pool of candidate routes (hypothetically suggested by the planner) from which the optimal subset is selected. This is a real application that we faced during our meetings with planners of the municipality of Rivera. Moreover, a solution was obtained using the proposed greedy constructive algorithm and a frequency optimization model; that solution is presented with gap values of around 20% and 30% with respect to on-board travel time and waiting time lower bounds respectively (Section 4.5.3).

In light of these observations, in our opinion we have proposed models and algorithms that exhibit an acceptable trade-off between realism and quality of approximation to the optimum and to real solutions, in relation to the state of the art.

6.2 The experiments and the application to real cases

In this section, we conclude about our experience in this thesis concerning the computational tests of the algorithms and their application to real cases.

6.2.1 Experiments

When performing numerical experiments, we faced the difficulty of lack of standard benchmark cases. The only case used by several authors and for which the corresponding data is available is the one proposed by Mandl [78]. Although that case is useful to make comparisons, it can not be used to validate a given model and/or algorithm, because it is not clear how it was constructed. In other words, we do not know which elements of the reality correspond to vertices and edges of its infrastructure graph.

The case of Rivera enabled us to experiment with a realistic scenario; the results produced by the models and algorithms are easy to interpret, by comparing with the routes and frequencies operating the public transportation system of the city. Although at first sight it could seem a very small-sized case, its complexity is sufficiently high to validate our work. We would like to experiment with a larger case, as it could be the public transportation system of Montevideo, capital city of Uruguay, which has about 1.5 million of inhabitants. In particular, we would want to confirm that the heuristics developed in this thesis are scalable, since they are based on subroutines of polynomial running time with respect to the size of the infrastructure graph and number of OD pairs. However, we estimated that the efforts needed to construct that case would be very high; in particular, origin-destination data at a relatively low level of aggregation was not available. We consider that high-quality data are mandatory in order to extract consistent conclusions from experiments with models and algorithms for the TNDP.

When comparing results with those published in the literature, we faced the difficulty related to the assignment model. In most cases, a consistent comparison requires to use the same hypothesis concerning the behavior of the users; this is hard to fulfill since the

assignment model is usually complex and there are many variations among the existing ones. Moreover, it is not easy to implement an existing assignment model, since many small details should be taken into account. In this thesis we used two different assignment models:

- Optimal strategies, proposed by Spiess and Florian [110]. It is used to construct the mathematical formulations proposed in Chapter 3, since it has an explicit formulation. To the best of our knowledge, no model or algorithm for the TNDP has previously used this model to represent the behavior of the users; [26] has used it in the context of frequency optimization. Our implementation of this model was validated by comparing with values taken from its original publication and by applying different implementations (the linear programming formulation as well as the label-setting algorithm proposed in [110]) to different cases, including those of Mandl and Rivera.

- The model model of Baaj and Mahmassani [8], that in general terms is similar to the optimal strategies assignment model. The main differences with respect to that model are: (i) it assumes different hypothesis concerning the behavior of the users with respect to transfers and (ii) it does not have an explicit mathematical formulation. By using this assignment model we were able to compare with results published in the literature. Our implementation of this model was validated by comparing with values taken from [9].

It is worth mentioning that the optimal strategies assignment model assumes that the users take into account a considerable amount of information and also that they are able to manage such information in order to determine the optimal strategy. In some real scenarios only certain OD pairs behave like this, while the rest apply a simpler approach. In principle, there are few different behaviors that could be modeled without changing the nature of the resulting formulation (which is linear). In case we need to model more specific behaviors, possibly the assignment model of Baaj and Mahmassani is more flexible in this sense.

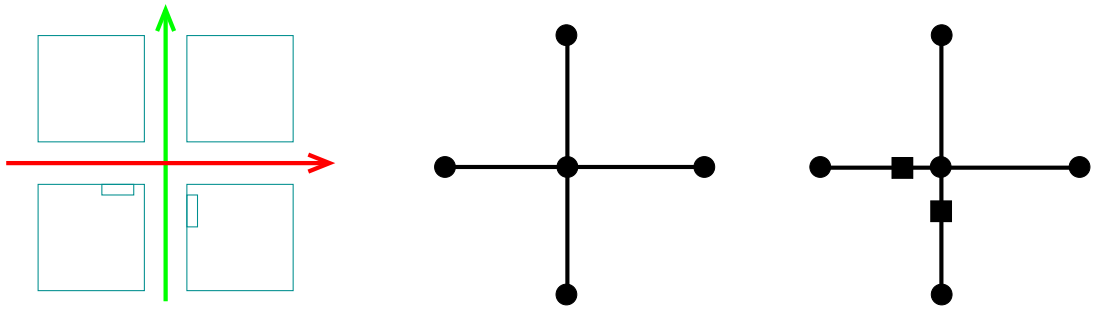
The assignment model is a crucial component of any model or algorithm for the TNDP. Since it represents the behavior of the passengers it strongly determines the realism of the resulting optimization model. Moreover, its computational implementation has strong influence on the efficiency of the solution algorithm.

6.2.2 Application to real cases

In this section we mention several issues that should be taken into account when concluding from numerical results obtained by our models and algorithms with the real case. Also we comment about the role of the methodologies proposed in this thesis, in the context of the planning of a real public transportation system.

The case of Rivera

The case was constructed by using the following approach: (i) walk arcs are not considered, (ii) every vertex is centroid, street and stop at the same time and (iii) the infrastructure



(a) Real system (b) Non detailed model (c) Detailed model

Figure 6.1: Modeling the bus stops in the TNDP

graph does not necessary follow the street network. That structure is consistent with the hypothesis of most models for the TNDP. However, it poses some difficulties when we want to evaluate the solution operating the public transportation system of the city; since an approximation of that solution is given as input to the models, the values obtained are subject to errors. This should be taken into account when comparing solutions generated by the optimization model against the real one, as it is done in Section 5.5.4. Moreover, it is worth mentioning that the modeling of the bus stops in the infrastructure graph strongly determines the validity of the assignment model. For example, a passenger waiting for any of the two lines represented in Figure 6.1(a) sees a single stop if we model the infrastructure graph as in Figure 6.1(b); however, in a more detailed model (Figure 6.1(c)) the passenger should first choose between these two lines and then wait at the corresponding stop. Observe that expressions for the waiting time (2.1) and the frequency-share rule (2.2) which are the core of most frequency-based assignment models, can not include both lines in such example. Thus, the modeling of the behavior of the users with respect to the bus stops is another potential source of error.

Either when constructing the case of Rivera as well as when performing the experiments with models and algorithms, we assumed that the users never perform transfers and that the OD matrix is fixed and independent of any set of routes and frequencies. These hypotheses (which can be reasonably assumed in this case) simplified our models, algorithms and analysis of results. However, in the more general case, a study of the valuation that users have of the different components of the travel (on-board travel and waiting time, transfers) with respect to different aspects like the quality of the infrastructure, would be needed. Also, the elasticity of the demand (in particular with respect to routes and frequencies) would need to be studied; its inclusion into the proposed models and algorithms poses and interesting challenge.

The evaluation of the solutions produced by our algorithms was done in terms of numerical values corresponding to measures of interest like travel time, fleet size and passenger flows representing occupancy levels of the buses. We did not perform any subjective evaluation based on a visual inspection of the resulting routes; we did not research about methodologies and we do not have sufficient local knowledge of the city of Rivera to do that. A possible comparison of the solutions obtained against the real one can be done in terms of the route structure, using the similarity measures proposed

in Section 4.5.2. The solutions proposed by our models and algorithms were not adopted in Rivera. The planners of the municipality recognized the difficulty of changing the structure of some historical routes, due to the social impact. But they identified the need of evaluating new alternatives to the current route structure and frequencies, looking for a possible reduction in the operation costs. In this context, they recognize the importance of having a tool that allows to design and evaluate changes, either proposed by researchers or consultants.

Our models and algorithms in a real application

The TNDP can arise at different stages of the planning of the public transportation system. At the strategic level we can mention several concrete applications: (i) the complete redesign of the routes, (ii) estimation of a value of fleet size necessary to bring the service with a given level, (iii) evaluation of alternative solutions with different trade-off between different objectives. When using the models proposed in this thesis for strategic planning, not all the constraints need to be included. Since perhaps an OD matrix corresponding to the whole day is used, including bus capacity constraints does not make sense, since OD values may be smoothed due to the large time horizon. On the other hand, demand covering and street capacity are important constraints that should be taken into account in this scenario; they can be verified without requiring information concerning real passenger flows over the lines. At the level of tactical planning, the models concerning the TNDP could be used to make adjustments to existing routes or to design feeder routes in a given region or neighborhood of the city. In this case, more detailed information is likely to be available, therefore a more detailed modeling is desirable. According to this, the bus capacity constraint can be included.

Although we stated some hypothesis that limit the scope of this thesis (Section 1.1), their relaxation is not contradictory with the models and algorithms developed. The routes and frequencies proposed by our algorithms can feed a system that models the interaction with other modes of transportation and even the elasticity of the demand. Thus, an iterative loop of route optimization and simulation of the dynamic of the city can be performed in order to obtain a realistic evaluation of the overall impact of a proposed route system. On the other hand, considering the effects of fares and advanced traveler information systems impacts in a high extent in the structure of the models and algorithms developed, specifically in the assignment sub-model.

6.3 Opinions and recommendations

To close this dissertation, in this section we mention some issues that in our opinion should be taken into account in future research concerning models and algorithms for the TNDP and their application to real cases.

We can conclude that either using exact or heuristic methods, it is hard to find an optimal solution and even to quantify the accuracy of any obtained solution; the gaps presented in this thesis as well as the ones presented in the existing literature concerning the TNDP are very high in comparison with other problems in the area of Operations Research. Taking into account these limitations, we note that when applying an optimization

model to the problem of route optimization in public transportation, generally it is sufficient to obtain a solution that is better (or non-dominated) with respect to the current one. In our opinion, exact methods based on explicit mathematical programming formulations to solve the TNDP considering waiting time and assignment to multiple routes, may be developed in the near future. However, to solve the more complex variant which includes transfer and bus capacity constraints (as proposed in Section 3.3), heuristic methods seem to be at this moment the only feasible approach for real cases. Concerning the development of heuristics for the TNDP, we note that both PIA and GRASP TNDP algorithms proposed on this thesis use problem knowledge intensively to obtain good solutions. This seems to be an appropriate approach since the problem includes complex constraints and an assignment sub-model. By contrast, a heuristic that performs blindly many moves (changes) to the solutions is likely to cause infeasibility and inefficiency due to repeated invocations to the assignment algorithm. Concerning the modeling of the problem we note that the hypothesis which excludes circular lines (which is present in most existing models for the TNDP) seems to be very restrictive, since such lines can be convenient for certain demand patterns in certain parts of the city.

When applying models and algorithms for the TNDP to real cases, special attention should be put in the modeling of the infrastructure and the demand. The zonal division of the city is a key aspect in the construction of the model. Observe that such a division should be made so that the lengths of the access/egress arcs that connect centroids with bus stops are reasonable. The way in which we construct the zonal division implicitly applies a criterion of geographical accessibility to the public transportation system. Most of the existing models for the TNDP and even the assignment model of Baaj and Mahmassani used by us do not consider the access/egress time; moreover, they assume implicitly that every vertex of the graph is centroid, street and stop at the same time. This is a reasonable assumption in order to simplify the models; we did the same when presenting our formulations and solution algorithms. However, when applying those models to real cases, a more detailed representation of the elements of the problem is likely to be made in order to construct a realistic scenario. Since demand data usually is presented at the level of zone centroid, we should include that type of vertex into the model; note that fixing the demand to bus stops is not a convenient alternative, since by doing this we are fixing part of the behavior of the users. That aspect of the problem is not commonly taken into account by researchers, and we consider that it is very important in order to obtain meaningful results. In the context of our thesis, the models and algorithms proposed can be easily extended to handle these elements; the software tool described in Appendix B allows to construct a case according to the discussion presented above.

Finally, all the discussions and conclusions written in this chapter are consistent with the general idea that in order to apply Operations Research techniques to a real problem, we have to deal with an appropriate combination of reasonably realistic modeling, efficient solution methods with some kind of quantification of accuracy, high quality real data (properly used and processed), critical interpretation of results and tools that allow to transfer such results to the application area. We tried to focus our efforts in this direction during the development of this thesis.

Appendix A

Real test case

In this appendix we describe the main aspects of the construction of the real test case used in Chapters 3, 4 and 5. That construction was done in the framework of a long-term project which includes this thesis. The web-site `http://www.fing.edu.uy/~mauttone/tndp` contains a description of the project as well as the data of the case. A more detailed description of the case is given in [81].

The case is related to a medium to small-sized city of 65,000 inhabitants in Uruguay, the city of Rivera. Public transportation has a strong presence in that city. When data gathering for this work was accomplished on August 2004, the system operated 13 bus lines with an average route length of 13.6 kilometers and an imposed duration (round-trip time) of 60 minutes each. Route headway ranges from 20 to 60 minutes. There are 11 lines which had forward and backward routes, whose structure differ slightly. The other 2 lines had a circular structure; each one of these lines had a single route, that is traversed in one direction. The inter-zonal demand had a radial pattern, being the city center the main attractor of trips. In a regular mid-week day, an average of 13,360 trips were performed using public transportation in Rivera. The users of this system have a negative perception of transfers and waiting time; in particular, the planners have designed the timetables of overlapping routes so as to reduce as much as possible that component of the overall travel time.

Data to construct a real test case for the TNDP is not easy to gather. Graph G can be generated from a representation of the street network; a software tool that manages geographical information can be helpful for this processing. However, to construct the origin-destination matrix D, a considerable amount of information should be compiled [99], to express the needs of public transportation between different points of the city. In the following we explain the construction of the case of Rivera, in terms of the zones, graph and demand.

Zones

The city was divided in zones. Each zone comprises approximately 4×4 blocks of 100 meters each. This size is intended to apply a criterion of geographical accessibility of the people to the public transportation system. We consider that 400 meters represent a maximum reasonable walk distance to access to a bus line passing through the zone. The

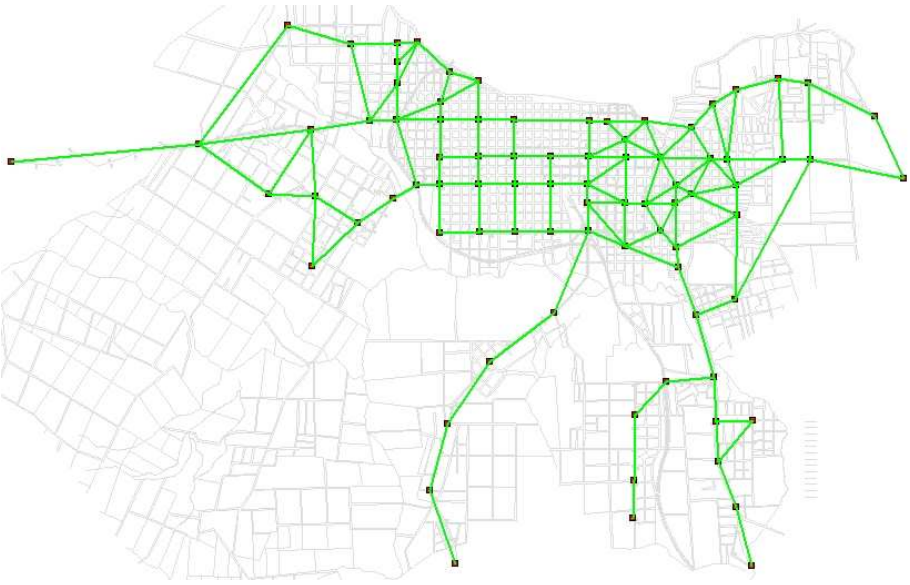


Figure A.1: Graph

demand produced (attracted) by a given zone is considered as covered when a line passes by any place in the street network inside the zone and inside the destination (origin) zone. Note that we are not considering access and egress times in this model.

Graph

The graph G is an abstraction of the real street network. Each vertex of G represents a zone of the zonal division; it is located over the intersection of streets that is nearest to the barycenter of the corresponding zone. An edge exists between two vertices in G if their corresponding zones are adjacent; its value of in-vehicle travel time is calculated from the distance of the shortest path in the street network between its extreme vertices and a bus commercial speed estimated in 13.6 kilometers per hour. The resulting graph of Rivera, constructed according to the explained procedure, has 84 vertices and 143 edges (Figure A.1).

Demand

Demand data were collected by means of a survey made on-board of lines operating on the public transportation system of Rivera. The methodology of the survey is based on the one proposed in [111]. A sample of 13 out of the 23 bus runs performed per hour by the lines of the system was selected; a run is a trip of a bus going over the route at a given time. For each one of these runs, origin and destination stops were recorded for every person using the bus. Data were collected on a time period of 12 hours. For each line, origin-destination counts were expanded according to the sample size. Line origin-destination matrices from the 13 lines were consolidated into a single system origin-destination matrix, containing average values in the 12 hours time horizon. This matrix is intended to represent the need for public transportation across all the operation period of the system. Given the size of

the considered time horizon, significant peaks of demand for some pairs of vertices in some time periods may be smoothed in the average; for this reason this matrix is not suitable for estimation of passengers flows. A final step in this data processing consist in transforming the origin-destination matrix from the level of bus stops to the level of zones. Note that by increasing the aggregation level of the demand data, we obtain an OD matrix that is independent on the stops and the lines. The resulting matrix constructed according to the explained procedure has 5% of non null elements (378 OD pairs). This matrix is based in observed values which depend on the particular lines operating on the system when data were collected. However we consider that for the city of Rivera, this matrix of observed trips is a good approximation to the matrix of desired trips, mainly due to (i) the highly captive characteristic of the users of the public transportation system of Rivera, (ii) the high spatial coverage of the city by the existing lines, (iii) transfers are rarely performed and (iv) the buses do not operate beyond their capacity (all the passengers that desire to board a given bus, can do it).

Appendix B

Software tool

In this appendix we describe the main features of the software tool developed to assist the research concerning models and algorithms for transit route optimization. The software is called *igoR-tp*, due to its name (in Spanish) "Interfase Gráfica para la Optimización de Recorridos en Transporte Público", translated to English as "Graphical Interface for Transit Route Optimization". igoR-tp comprises three modules that implement features related to different activities:

- **Construction.** This module is responsible for constructing a test case. It allows to input data related to the street network, demand and zones. It generates the infrastructure graph and the origin-destination matrix.
- **Experimentation.** Once a case is constructed, this module enables to run algorithms that evaluate and optimize sets of routes and frequencies. The module calls the algorithms and displays the results. Also, it allows to create manually routes and set their frequencies or to modify existing ones.
- **Algorithms.** Since new algorithms may arise during the research, this module allows to integrate them into a library.

igoR-tp was specified in the context of this thesis and implemented by two undergraduate projects [3, 54] and a research and development project [114]. The software is built upon the MapWinGIS ActiveX Control [79], an open source project leaded by the Geospatial Software Lab of the Idaho State University. igoR-tp works with georeferenced data in the ESRI [39] shape format. The features included in the tool were specially developed for the purposes of the research on models and algorithms for transit network design. In the following we describe the main features of each module of igoR-tp.

Construction

In order to construct a case we should provide a database of georeferenced demand points; those points can represent households, bus stops, blocks or any place where we have data related to generation of trips for public transportation. Also we should provide the street network of the city, in shape format; it can be given directly as a network (street intersections and segments) or alternatively as a set of polylines, each one representing an entire

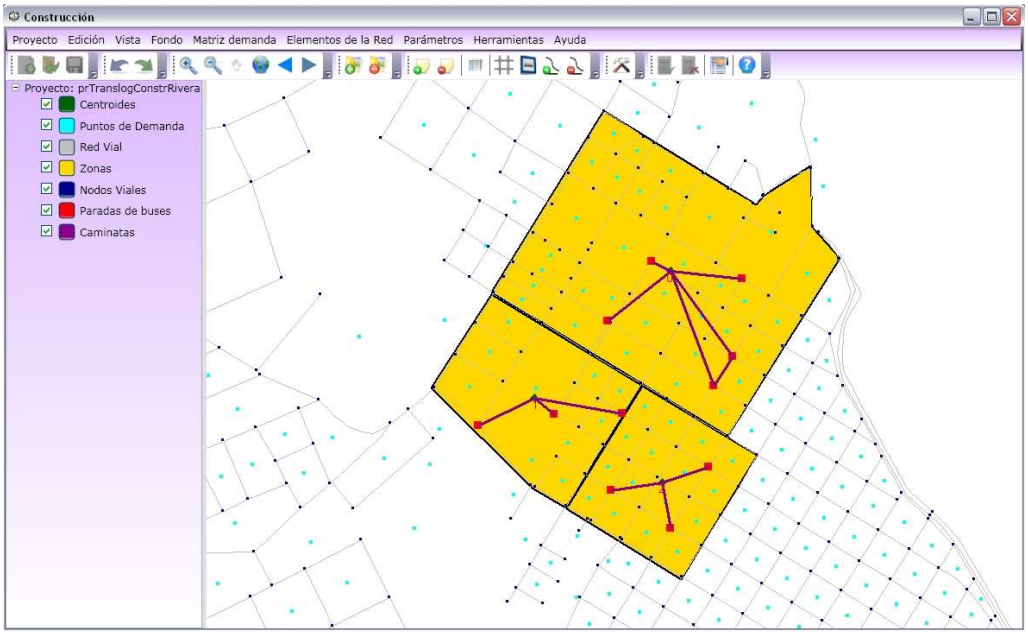


Figure B.1: Construction module

street. Street nodes and travel edges of the infrastructure graph are generated automatically from the street network. Some street nodes can be also of stop type: this should be specified by the user. Then, zones should be drawn; centroid vertices are generated automatically by the software. Walk edges between centroids and stops should be specified (Figure B.1). The full level of detail of the model as presented in Section 2.1 can not be implemented in this version of the software: the street direction is not considered and stop vertices can not be independent of street vertices. Finally, the origin-destination matrix is generated by summing the values of the demand points located inside every zone; note that other methods for processing the demand data could be implemented. Once the case is created, the OD pairs can be explored by navigating a list sorted by demand value, identifying the origin and destination vertices on the map.

Experimentation

In this module we can load a case created with the construction module to experiment with it. A solution is a set of routes with frequencies from the viewpoint of this module; the routes are defined over the infrastructure graph according to the definitions given in Section 2.1.2. Two types of algorithms can be invoked from the experimentation module: (i) evaluation, which computes measures of interest like travel time and occupancy level, from an existing solution and (ii) optimization, which generates a solution or several ones. Since different evaluation and optimization algorithms may be available, we should select one of them; then, we should enter the parameters according to the selected algorithm. The module manages different solutions, which are input or output of the algorithms. The routes of a solution can be created or modified manually; they can be undirected, directed or circular (Figure B.2). Some features related to displaying information are implemented,

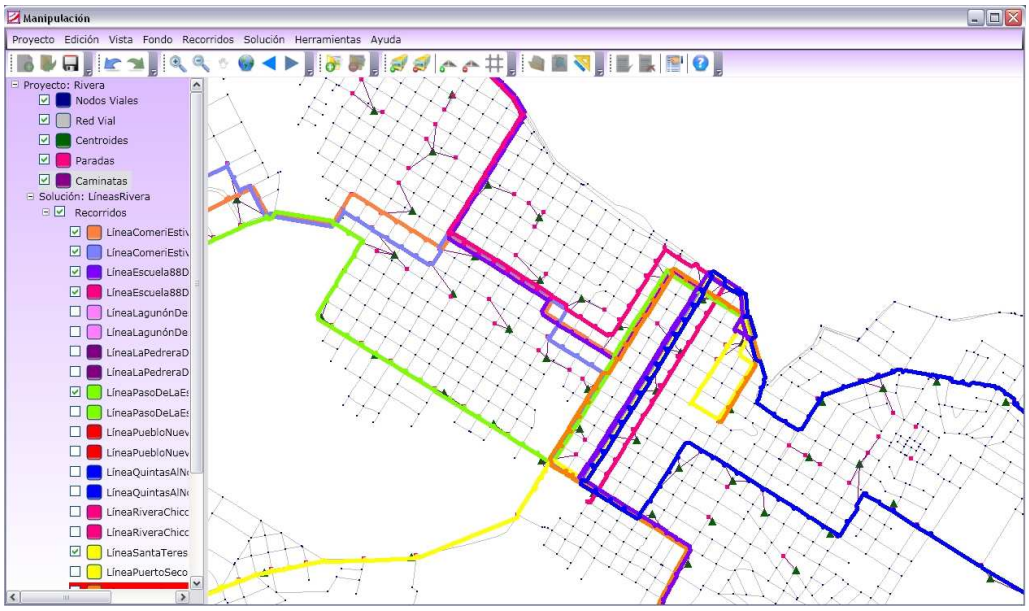


Figure B.2: Experimentation module

notably the visualization of overlapping routes, which is one of the most difficult issues concerning the visualization of solutions for the TNDP.

Algorithms

This is a small module which allows to add new evaluation and optimization algorithms, to the library of algorithms that can be called from the experimentation module. In order to add a new algorithm, we should provide its executable program and specify the list of parameters that it takes as input. The executable should accept a pre-specified set of parameters in order to be called, which are common to all algorithms (among them, the infrastructure graph and the origin-destination matrix).

igoR-tp has been tested with different versions of the case of the city or Rivera. The software proved to be capable of managing all the information required to work with a case relative to a small-sized city. We also tested the software with information related to Montevideo, capital city of Uruguay, whose population is 1,500,000 approximately. Although we did not intend to construct a test case related to that city (in particular, demand data is not available), we processed its street network. Once we created the infrastructure graph, we identified a degradation in the times needed to build the zones, with respect to the times experienced when working with Rivera. Some work should be done in order to extend the range of applications of igorR-tp: (i) study (and possibly improve) the performance of the system when working with larger cases, (ii) implement the full level of detail of the infrastructure graph according to definitions given in Section 2.1 and (iii) include mechanisms which allow to incorporate other methods for processing the demand data.

Bibliography

[1] J. Agrawal and M. Tom. Transit route network design using parallel genetic algorithm. *Journal of Computing in Civil Engineering*, 18(3):248–256, 2004.

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows*. Prentice-Hall, 1993.

[3] P. Aldaz and G. De Leon. *Simulador de transporte público urbano colectivo*. Undergraduate project in Computer Engineering (ongoing), Universidad de la República, 2010.

[4] R. Alvarez, M. Martínez, and A. Mauttone. Heurística de búsqueda de entorno variable para el problema de ruteo de transporte público urbano. In *42º Simpósio Brasileiro de Pesquisa Operacional*, Bento Gonçalves, Brazil, 2010.

[5] A. Américo, F. Martínez, A. Mauttone, and M.E. Urquhart. Multi-objective evolutionary algorithm for the transit network design problem. In *VI International Conference on Operational Research for Development*, Fortaleza, Brazil, 2007.

[6] J. An, J. Teng, and L. Meng. A BRT network route design model. In *11th International IEEE Conference on Intelligent Transportation Systems*, pages 734–741, Beijing, China, 2008.

[7] K. W. Axhausen and R. L. Smith. Evaluation of heuristic transit network optimization algorithms. *Transportation Research Record*, (976):7–20, 1984.

[8] M. H. Baaj and H. S. Mahmassani. TRUST: A LISP program for the analysis of transit route configurations. *Transportation Research Record*, (1283):125–135, 1990.

[9] M. H. Baaj and H. S. Mahmassani. An AI-based approach for transit route system planning and design. *Journal of Advanced Transportation*, 25(2):187–210, 1991.

[10] M. H. Baaj and H. S. Mahmassani. Hybrid route generation heuristic algorithm for the design of transit networks. *Transportation Research C*, 3(1):31–50, 1995.

[11] G. Baldoquín. Approximate solution of an extended 0/1 knapsack problem using grasp. In *XI Congreso Latino-Iberoamericano de Investigación de Operaciones*, Concepción, Chile, 2002.

[12] J.F. Bard. *Practical Bilevel Optimization*. Kluwer, 1998.

[13] D. Bertsimas and J.N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[14] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.

[15] R. Borndörfer, M. Grötschel, and M. Pfetsch. A column-generation approach to line planning in public transport. *Transportation Science*, 41(1):123–132, 2007.

[16] B. Bouzaïene-Ayari, M. Gendreau, and S. Nguyen. Modeling bus stops in transit networks: A survey and new formulations. *Transportation Science*, 35(3):304–321, 2001.

[17] A. Ceder. Bus frequency determination using passenger count data. *Transportation Research A*, 18(5-6):439–453, 1984.

[18] A. Ceder and N. H. M. Wilson. Bus network design. *Transportation Research B*, 20(4):331–344, 1986.

[19] M. Cepeda, R. Cominetti, and M. Florian. A frequency-based assignment model for congested transit networks with strict capacity constraints: characterization and computation of equilibria. *Transportation Research B*, 40(6):437–459, 2006.

[20] P. Chakroborty. Genetic algorithms for optimal urban transit network design. *Computer-Aided Civil and Infrastructure Engineering*, 18(3):184–200, 2003.

[21] P. Chakroborty and T. Dwivedi. Optimal route network design for transit systems using genetic algorithms. *Engineering Optimization*, 34(1):83–100, 2002.

[22] C. Chriqui and P. Robillard. Common bus lines. *Transportation Science*, 9(2):115–121, 1975.

[23] E. Cipriani, S. Gori, and M. Petrelli. Transit network design: A procedure and an application to a large urban area. *Transportation Research C*, 2010. In press.

[24] C.A. Coello. An updated survey of ga-based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2):109–143, 2000.

[25] B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256, 2007.

[26] I. Constantin and M. Florian. Optimizing frequencies in a transit network: a non-linear bi-level programming approach. *International Transactions in Operational Research*, 2(2):149–164, 1995.

[27] J.R. Correa, A.S. Schulz, and N.E. Stier-Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29(4):961–976, 2004.

[28] G.B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[29] J. de Cea and E. Fernández. Transit assignment to minimal routes: An efficient new algorithm. *Traffic Engineering and Control*, 30(10):491–494, 1989.

[30] J. de Cea and E. Fernández. Transit assignment for congested public transport systems: An equilibrium model. *Transportation Science*, 27(2):133–147, 1993.

[31] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons, 2001.

[32] S. Dempe. Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. *Optimization*, 52(3):333–359, 2003.

[33] G. Desaulniers and M. D. Hickman. Public transit. In G. Laporte and C. Barnhart, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 69–127. Elsevier, Amsterdam, 2007.

[34] R.B. Dial. Transit pathfinder algorithm. *Highway Research Record*, 205:67–85, 1967.

[35] D. Dubois, G. Bel, and M. Llibre. A set of methods in transportation network synthesis and analysis. *Journal of the Operational Research Society*, 30(9):797–808, 1979.

[36] M. Ehrgott and X. Gandibleux. An annotated bibliography of multiobjective combinatorial optimization. Technical Report 62/2000, Fachbereich Mathematik - Universitat Kaiserslautern, 2000.

[37] M. Ehrgott and X. Gandibleux. Multiobjective combinatorial optimization. In M. Ehrgott and X. Gandibleux, editors, *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*, International Series in Operations Research ans Management Science, pages 369–444. Kluwer, 2002.

[38] M. Ehrgott and X. Gandibleux. Approximative solution methods for multiobjective combinatorial optimization. *TOP*, 12(1):1–89, 2004.

[39] ESRI. http://www.esri.com/.

[40] L. Fan and C. Mumford. A metaheuristic approach to the urban transit routing problem. *Journal of Heuristics*, 16(3):353–372, 2010.

[41] W. Fan and R. B. Machemehl. Optimal transit route network design problem: Algorithms, implementations, and numerical results. Technical Report 167244-1, University of Texas, 2004.

[42] W. Fan and R. B. Machemehl. Optimal transit route network design problem with variable transit demand: Genetic algorithm approach. *Journal of Transportation Engineering*, 132(1):40–51, 2006.

[43] W. Fan and R. B. Machemehl. Using a simulated annealing algorithm to solve the transit route network design problem. *Journal of Transportation Engineering*, 132(2):122–132, 2006.

[44] W. Fan and R. B. Machemehl. A tabu search based heuristic method for the transit route network design problem. In M. Hickman, P. Mirchandani, and S. Voß, editors, *Computer-aided Systems in Public Transport*, Lecture Notes in Economics and Mathematical Systems, pages 387–408. Springer, 2008.

[45] T. Feo and M. Resende. Greedy randomized adaptative search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[46] E. Fernández, J. de Cea, and I. Norambuena. Una metodología para el diseño topológico de sistemas de transporte público urbano de pasajeros. In *XI Congreso Chileno de Ingeniería de Transporte*, Santiago, Chile, 2003.

[47] J.E. Fernández, J. de Cea, and R.H. Malbran. Demand responsive urban public transport system design: Methodology and application. *Transportation Research A*, 42(7):951–972, 2008.

[48] Fernández y de Cea Ingenieros Limitada. http://www.fdcconsult.com/.

[49] J. Fortuny-Amat and B. McCarl. A representation and economic interpretation of a two-level programming problem. *Journal of the Operational Research Society*, 32:783–792, 1981.

[50] R. Freling, D. Huisman, and A. Wagelmans. Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling*, 6(1):63–85, 2003.

[51] X. Gandibleux, D. Vancoppenolle, and D. Tuyttens. A first making use of GRASP for solving MOCO problems. Technical report, University of Valenciennes, 1998.

[52] Z. Gao, H. Sun, and L. L. Shan. A continuous equilibrium network design model and algorithm for transit systems. *Transportation Research B*, 38(3):235–250, 2004.

[53] M. Garey and D. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[54] D. Gawenda and H. Martínez. *Interfaz para herramienta de planificación de recorridos para transporte público*. Undergraduate project in Computer Engineering, Universidad de la República, 2009.

[55] M. Gendreau. *Etude approfondie d'un modèle d'équilibre pour l'affectation de passagers dans les réseaux de transports en commun*. Ph.d. thesis, Université de Montréal, 1984. Publication CRT-384.

[56] F.W. Glover and G.A. Kochenberger, editors. *Handbook of Metaheuristics*. International Series in Operations Research and Management Science. Springer, 2003.

[57] F.W. Glover and M. Laguna. *Tabu Search*. Springer, 1998.

[58] J.F. Guan, H. Yang, and S.C. Wirasinghe. Simultaneous optimization of transit line configuration and passenger line assignment. *Transportation Research B*, 40(10):885–902, 2006.

[59] V. Guihaire and J.-K. Hao. Transit network design and scheduling: A global review. *Transportation Research A*, 42(10):1251–1273, 2008.

[60] A. F. Han and N.H. M. Wilson. The allocation of buses in heavily utilized networks with overlapping routes. *Transportation Research B*, 13(3):221–232, 1982.

[61] D. Hasselström. *Public transportation planning - Mathematical programming approach*. Phd thesis, University of Gothenburg, 1981.

[62] A.J. Higgins, S. Hajkowicz, and E. Bui. A multi-objective model for environmental investment decision making. *Computers & Operations Research*, 35(1):253–266, 2008.

[63] A. Ibeas, L. dell'Olio, B. Alonso, and O. Sainz. Optimizing bus stop spacing in urban areas. *Transportation Research E*, 46(3):446–458, 2010.

[64] INRO. http://www.inro.ca.

[65] Y. Israeli and A. Ceder. Transit route design using scheduling and multiobjective programming techniques. In J. Daduna, I. Branco, and J. P. Paixão, editors, *Computer-Aided Transit Scheduling: Proceedings of the Sixth International Workshop on Computer-Aided Scheduling of Public Transport*, Lecture Notes in Economics and Mathematical Systems, pages 56–75. Springer, 1995.

[66] A. Jaszkiewicz. Evaluation of multiple objective metaheuristics. In X. Gandibleux, M. Sevaux, and K. Swrensen, editors, *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, 2004.

[67] B.Y. Kara and V. Verter. Designing a road network for hazardous materials transportation. *Transportation Science*, 38(2):188–196, 2004.

[68] K. Kepaptsoglou and M. Karlaftis. Transit route network design problem: Review. *Journal of Transportation Engineering*, 135(8):491–505, 2009.

[69] W.H.K. Lam and M.G.H. Bell, editors. *Advanced Modeling for Transit Operations and Service Planning*. Elsevier, Oxford, 2003.

[70] W. Lampkin and P. D. Saalmans. The design of routes, service frequencies, and schedules for a municipal bus undertaking: A case study. *Operational Research Quarterly*, 18(4):375–397, 1967.

[71] H. Larrain and J.C. Muñoz. Public transit corridor assignment assuming congestion due to passenger boarding and alighting. *Networks and Spatial Economics*, 8(2-3):241–256, 2008.

[72] Y.J. Lee and V. Vuchic. Transit network design with variable demand. *Journal of Transportation Engineering*, 131(1):1–10, 2005.

[73] C. Leiva, J.C. Muñoz, R. Giesen, and H. Larrain. Design of limited-stop services for an urban bus corridor with capacity constraints. *Transportation Research B*, 44(10):1186–1201, 2010.

[74] H. Li and D. Landa-Silva. An elitist grasp metaheuristic for the multi-objective quadratic assignment problem. In M. Ehrgott, C.M. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, editors, *Evolutionary Multi-Criterion Optimization*, volume 5467 of *Lecture Notes in Computer Science*, pages 481–494. Springer, 2009.

[75] OR Library. http://people.brunel.ac.uk/ mastjjb/jeb/info.html.

[76] T. L. Magnanti and R. T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

[77] C. E. Mandl. Evaluation and optimization of urban public transportation networks. *European Journal of Operational Research*, 5(6):396–404, 1980.

[78] C.E. Mandl. *Applied Network Optimization*. Academic Press, 1980.

[79] MapWindow. http://www.mapwindow.org/.

[80] A. Marín. An extension to rapid transit network design problem. *TOP*, 15:231–241, 2007.

[81] A. Mauttone. *Optimización de recorridos y frecuencias en sistemas de transporte público urbano colectivo*. Master thesis on Informatics, Universidad de la República, 2005.

[82] A. Mauttone. Formulación de programación matemática para el problema de optimización de recorridos y frecuencias en sistemas de transporte público. Technical Report RT 09-14, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, 2009.

[83] A. Mauttone, R. Giesen, and M.E. Urquhart. Transit network design problem: A mathematical formulation and heuristic solution. In *Transportation and Logistics Workshop*, Reñaca, Chile, 2009.

[84] A. Mauttone, M. Labbé, and R. M. V. Figueiredo. A tabu search approach to solve a network design problem with user-optimal flows. In *VI ALIO/EURO Workshop on Applied Combinatorial Optimization*, Buenos Aires, Argentina, 2008.

[85] A. Mauttone and M. Urquhart. A route set construction algorithm for the transit network design problem. *Computers & Operations Research*, 36(8):2440–2449, 2009.

[86] A. Mauttone and M.E. Urquhart. GRASP para el diseño de recorridos en transporte público. In *XII Congreso Latino-Iberoamericano de Investigación de Operaciones*, La Habana, Cuba, 2004.

[87] A. Mauttone and M.E. Urquhart. A multi-objective metaheuristic approach for the transit network design problem. In *10th International Conference on Computer-Aided Scheduling of Public Transport*, Leeds, United Kingdom, 2006.

[88] A. Mauttone and M.E. Urquhart. Una heurística basada en memoria para el problema del diseño de recorridos en transporte público urbano. In *XIII Congreso Latino Iberoamericano de Investigación Operativa*, Montevideo, Uruguay, 2006.

[89] A. Mauttone and M.E. Urquhart. Optimización multi-objetivo de recorridos y frecuencias en transporte público aplicado a un caso de estudio real. In *XIII Congreso Chileno de Ingeniería de Transporte*, Santiago, Chile, 2007.

[90] A. Mauttone and M.E. Urquhart. A multi-objective metaheuristic approach for the transit network design problem. *Public Transport*, 1(4):253–273, 2009.

[91] M. Michaelis and A. Schöbel. Integrating line planning, timetabling, and vehicle scheduling: a customer-oriented heuristic. *Public Transport*, 1(3):211–232, 2009.

[92] A. Migdalas. Bilevel programming in traffic planning: models, methods and challenge. *Journal of Global Optimization*, 7:381–405, 1995.

[93] J.C. Muñoz and R. Giesen. Optimization of public transportation systems. In J.J. Cochran, editor, *Encyclopedia of Operations Research and Management Science*, volume 6, pages 3886–3896. 2010.

[94] S. Ngamchai and D. Lovell. Optimal time transfer in bus transit route network design using a genetic algorithm. *Journal of Transportation Engineering*, 129(5):510–521, 2003.

[95] S. Nguyen and S. Pallottino. Equilibrium traffic assignment for large scale transit networks. *European Journal of Operational Research*, 37(2):176–186, 1988.

[96] O.A. Nielsen. A stochastic transit assignment model considering differences in passengers utility functions. *Transportation Research B*, 34(5):377–402, 2000.

[97] A. Nuzzolo. Transit path choice and assignment model approaches. In W. Lam and M. Bell, editors, *Advanced Modeling for Transit Operations and Service Planning*, pages 93–124. Elsevier, Oxford, 2003.

[98] A. Odoni, J.-M. Rousseau, and N.H.M. Wilson. Models in urban and air transportation. In S. M. Pollock, M. H. Rothkopf, and A. Barnett, editors, *Operations Research and the Public Sector*, volume 6 of *Handbooks in Operations Research and Management Science*, pages 107–150. Elsevier, 1994.

[99] J. de D. Ortúzar and L. Willumnsen. *Modelling Transport*. John Wiley and Sons, 1996.

[100] J. Pacheco, A. Alvarez, S. Casado, and J.L. González-Velarde. A tabu search approach to an urban transport problem in northern spain. *Computers & Operations Research*, 36(3):967–979, 2009.

[101] S. B. Pattnaik, S. Mohan, and V. M. Tom. Urban bus transit route network design using genetic algorithm. *Journal of Transportation Engineering*, 124(4):368–375, 1998.

[102] PTV. http://www.ptvag.com/.

[103] K. V. K. Rao, S. Muralidhar, and S. L. Dhingra. Public transport routing and scheduling using genetic algorithms. In *8th International Conference on Computer Aided Scheduling of Public Transport*, Berlin, Germany, 2000.

[104] M. Resende and C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 219–249. Kluwer Academic Publishers, 2003.

[105] S. Sayin and P. Kouvelis. The multiobjective discrete optimization problem: A weighted min-max two-stage optimization approach and a bicriteria algorithm. *Management Science*, 51(10):1572–1581, 2005.

[106] A. Schöbel. Line planning in public transportation: mathematical programming approaches. Technical Report 2009-20, Institut für Numerische und Angewandte Mathematik, Georg-August Universität Göttingen, 2009.

[107] A. Schöbel and S. Scholl. Line planning with minimal traveling time. In L.G. Kroon and R.H. Möhring, editors, *5th Workshop on Algorithmic Methods and Models for Optimization of Railways*, 2005.

[108] Y. Sheffi. *Urban Transportation Networks*. Prentice-Hall, 1985.

[109] L. A. Silman, Z. Barziliy, and U. Passy. Planning the route system for urban buses. *Computers & Operations Research*, 1(2):201–211, 1974.

[110] H. Spiess and M. Florian. Optimal strategies: a new assignment model for transit networks. *Transportation Research B*, 23(2):83–102, 1989.

[111] P. R. Stopher, L. Shillito, D.T. Grober, and H.M.A. Stopher. On-board bus surveys: No questions asked. *Transportation Research Record*, (1085):50–57, 1986.

[112] W.Y. Szeto and Y. Wu. A simultaneous bus route design and frequency setting problem for tin shui wai, hong kong. *European Journal of Operational Research*, 209(2):141–155, 2011.

[113] V. M. Tom and S. Mohan. Transit route network design using frequency coded genetic algorithm. *Journal of Transportation Engineering*, 129(2):186–195, 2003.

[114] M. Urquhart. Planificación de líneas de ómnibus en el transporte público urbano colectivo. Final report Project 48/02, Programa de Desarrollo Tecnológico, 2009.

[115] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[116] R. van Nes. *Optimal Stop and Line Spacing for Urban Public Transport Networks*. Delft University Press, 2000.

[117] D.S. Vianna and J.E.C. Arroyo. A GRASP algorithm for the multi-objective knapsack problem. In *24th International Conference of the Chilean Computer Science Society*, pages 69–75, Chile, 2004.

[118] L. Vicente, G. Savard, and J. Judice. Discrete linear bilevel programming problem. *Journal of Optimization Theory and Applications*, 89(3):597–614, 1996.

[119] J.L. Walteros and A.L. Medaglia. Hybrid algorithm for route design on bus rapid transit systems. Technical Report COPA 2010 - 2, Centro para la Optimización y Probabilidad Aplicada, Universidad de los Andes, 2010. http://hdl.handle.net/1992/1128.

[120] Q. K. Wan and H. K. Lo. A mixed integer formulation for multiple-route transit network design. *Journal of Mathematical Modelling and Algorithms*, 2(4):299–308, 2003.

[121] Q.K. Wan and H.K. Lo. Congested multimodal transit network design. *Public Transport*, 1(3):233–251, 2009.

[122] J.Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1972.

[123] F. Zhao and X. Zeng. Optimization of transit route network, vehicle headways and timetables for large-scale transit networks. *European Journal of Operational Research*, 186:841–855, 2008.

[124] J. Zhou and W.H.K. Lam. Models for optimizing transit fares. In W.H.K. Lam and M.G.H. Bell, editors, *Advanced Modeling for Transit Operations and Service Planning*, pages 315–345. Elsevier, Oxford, 2003.