Programa de Doctorado "Matemáticas"

PhD Dissertation

Mathematical Models for the Design and Planning for Transportation on Demand in Urban Logistics Networks

Author
Miguel A. Pozo Montaño

Supervisor
Prof. Dr. *Francisco A. Ortega Riejos*

Supervisor
Prof. Dr. *Justo Puerto Albandoz*

May 13, 2015

Abstract

The freight-transport industry has made enormous progress in the development and application of logistics techniques that has transformed its operation, giving raise to impressive productivity gains and improved responsiveness to its consumers. While the separation of passenger and freight traffic is a relatively new concept in historic terms, recent approaches point out that most freight-logistics techniques are transferable to the passenger-transport industry. In this sense, passenger logistics can be understood as the application of logistics techniques in urban contexts to the passenger-transport industry. The design of an urban logistic network integrates decisions about the emplacement, number and capacities of the facilities that will be located, the flows between them, demand patterns and cost structures that will validate the profitability of the process. This strategic decision settles conditions and constraints of latter tactical and operative decisions. In addition, different criteria are involved during the whole process so, in general terms, it is essential an exhaustive analysis, from the mathematical point of view, of the decision problem. The optimization models resulting from this analysis require techniques and mathematical algorithms in constant development and evolution. Such methods demand more and more a higher number of interrelated elements due to the increase of scale used in the current logistics and transportation problems.

This PhD dissertation explores different topics related to Mathematical models for the design and planning of transportation on demand in urban logistics networks. The contributions are divided into six main chapters since and, in addition, Chapter 0 offers a basic background for the contents that are presented in the remaining six chapters.

Chapter 1 deals with the Transit Network Timetabling and Scheduling Problem (TNTSP) in a public transit line. The TNTSP aims at determining optimal timetables for each line in a transit network by establishing departure and arrival times of each vehicle at each station. We assume that customers know departure times of line runs offered by the system. However, each user, traveling later of before their desired travel time, will give rise to an inconvenience cost, or a penalty cost if that user cannot be served according to the scheduled timetable. The provided formulation allocates each user to the best possible timetable considering capacity constraints. The problem is formulated using a p-median based approach and solved using a clustering technique. Computational results that show useful applications of this methodology are also included.

Chapter 2 deals with the TNTSP in a public transit network integrating in the model the passengers' routings. The current models for planning timetables and vehicle schedules use the knowledge of passengers' routings from the results of a previous phase. However, the actual route a passenger will take strongly depends on the timetable, which is not yet known a priori. The provided formulation guarantees that each user is allocated to the best possible timetable ensuring capacity constraints.

Chapter 3 deals with the rescheduling problem in a transit line that has suffered a fleet size reduction. We present different modelling possibilities depending on the assumptions that need to be included in the modelization and we show that the problem can be solved rapidly by using a constrained max-cost-flow problem whose coefficient matrix we prove is totally unimodular. We test our results in a

testbed of random instances outperforming previous results in the literature. An experimental study, based on a line segment of the Madrid Regional Railway network, shows that the proposed approach provides optimal reassignment decisions within computation times compatible with real-time use.

In Chapter 4 we discuss the multi-criteria p-facility median location problem on networks with positive and negative weights. We assume that the demand is located at the nodes and can be different for each criterion under consideration. The goal is to obtain the set of Pareto-optimal locations in the graph and the corresponding set of non-dominated objective values. To that end, we first characterize the linearity domains of the distance functions on the graph and compute the image of each linearity domain in the objective space. The lower envelope of a transformation of all these images then gives us the set of all non-dominated points in the objective space and its preimage corresponds to the set of all Pareto-optimal solutions on the graph. For the bicriteria 2-facility case we present a low order polynomial time algorithm. Also for the general case we propose an efficient algorithm, which is polynomial if the number of facilities and criteria is fixed.

In Chapter 5, Ordered Weighted Average optimization problems are studied from a modeling point of view. Alternative integer programming formulations for such problems are presented and their respective domains studied and compared. In addition, their associated polyhedra are studied and some families of facets and new families of valid inequalities presented. The proposed formulations are particularized for two well-known combinatorial optimization problems, namely, shortest path and minimum cost perfect matching, and the results of computational experiments presented and analyzed. These results indicate that the new formulations reinforced with appropriate constraints can be effective for efficiently solving medium to large size instances.

In Chapter 6, the multiobjective Minimum cost Spanning Tree Problem (MST) is studied from a modeling point of view. In particular, we use the ordered median objective function as an averaging operator to aggregate the vector of objective values of feasible solutions. This leads to the Ordered Weighted Average Spanning Tree Problem (OWASTP), which we study in this work. To solve the problem, we propose different integer programming formulations based in the most relevant MST formulations and in a new one. We analyze several enhancements for these formulations and we test their performance over a testbed of random instances. Finally we show that an appropriate choice will allow us to solve larger instances with more objectives than those previously solved in the literature.

Index

Preface

The freight-transport industry has made enormous progress in the development and application of logistics techniques that has transformed its operation, giving raise to impressive productivity gains and improved responsiveness to its consumers. While the separation of passenger and freight traffic is a relatively new concept in historic terms, recent approaches point out that most freight-logistics techniques are transferable to the passenger-transport industry. In this sense, passenger logistics can be understood as the application of logistics techniques in urban contexts to the passenger-transport industry. The design of an urban logistic network integrates decisions about the emplacement, number and capacities of the facilities that will be located, the flows between them, demand patterns and cost structures that will validate the profitability of the process. This strategic decision settles conditions and constraints of latter tactical and operative decisions. In addition, different criteria are involved during the whole process so, in general terms, it is essential an exhaustive analysis, from the mathematical point of view, of the decision problem. The optimization models resulting from this analysis require techniques and mathematical algorithms in constant development and evolution. Such methods demand more and more a higher number of interrelated elements due to the increase of scale used in the current logistics and transportation problems.

Transportation on Demand (TOD) copes with a set of transportation requests that are formulated between pickup and delivery points (origins and destinations respectively) and must be served by vehicles of a given capacity. In addition, time windows are usually specified in both pickup and delivery locations. TOD has been usually applied to the transportation of elderly and disabled as well as to transportation in rural areas. However, this perspective of social and economical interests is migrating to a more general passenger logistics framework. For example, the recent development of smartphones allows to collect personal and microscopic information about the preferences and habits of each passenger that can be later analyzed with Big Data tools. In the recent years TOD systems have gained popularity and interesting methodologies have been developed. The high cost of door-to-door transportation and the lack of flexibility of fixed route systems suggest the implementation of intermediate systems that allow serving transportation requests in centralized nodes. A relevant tactical decision consists of locating transfer nodes that let moving passengers between vehicles of different routes shortening the routes and decreasing transportation costs.

Even when it is possible to obtain high quality solutions for each sub-problem of a logistic network, sequential approaches lead to solutions that do not necessarily guarantee a cohesive solution for the planning problem as a whole. To overcome this situation, recent studies address the integration of two or more sub-problems at a time. There are two common approaches to integrate two or more problems: solving partial integrated formulations that consider some characteristics of one problem while taking decisions of other subproblems and defining complete integrated formulations and/or solution approaches that jointly determine the decisions of the problems. Since complete integrations consider all degrees of freedom of each sub-problem, they are more difficult to define and handle.

From another perspective, failures and interruptions are not usually considered in the planning phase of a logistic network so, even if correctly designed, it can be affected by big perturbations after an unexpected event (interruptions, accidents, etc) or during time intervals with a peak of demand. The existing uncertainty in many of the elements that take part in the problem, as the nature of the demand or supplies, the variability of transportation and operation costs, the reliability of the network, etc, suggest to propose robust solutions including random elements to the optimization problem or protecting the system against adverse or risky situations. A first possibility consists of designing auction protocols for each possible adverse scenario. A more conservative perspective searches robust solutions that fit in all scenarios. Recent developments point at providing on-line solutions as a response to disruptions within minutes.

The strategic planning is the starting phase of analysis in a Logistic Network and, therefore, location models play a key role at this stage. Broadly speaking, these models consider regions where a set of clients is established and their demand require to be satisfied with the location of one or several types of facilities. Besides, many real-world applications are concerned with finding an optimal location for one or more new facilities on a network (road network, power grid, etc.) minimizing a function of the distances between these facilities and a given set of existing facilities (clients, demand points), where the latter typically coincide with vertices. This is the case for example of the location of centralized nodes that could be used to serve transportation requests. The majority of research focuses on the minimization of a single objective function that is increasing with distance. However, in the process of locating a new facility usually more than one decision maker is involved. This is due to the fact that often the cost incurred with the decision is relatively high. Furthermore, different decision makers may (or will) have different (conflicting) objectives. An additional difficulty is that we are usually dealing with conflicting criteria and a single optimal solution does not always exist (which would be an optimal solution for each of the criteria). Despite its intrinsic interest, the multicriteria multi-facility location problem on networks has received little or none attention in the literature.

Multiobjective combinatorial optimization deals with problems considering more than one viewpoint or scenario. The problem of aggregating multiple criteria to obtain a globalizing objective function is of special interest when the number of Pareto solutions becomes considerably large or when a single, meaningful solution is required. For these reasons, more involved decision criteria have been proposed in the field of multicriteria decision making. These include objectives focusing on one particular compromise solution, which, for tractability and decision theoretic reasons, seem to be better suited when an appropriate aggregation operator is available. Ordered Weighted Average or Ordered Median operators are very useful when preferential information is available and objectives are comparable since they assign importance weights not to specific objectives but to their sorted values.

Contributions

This PhD dissertation is divided into six main chapters. Chapter 0 offers a basic background for the contents that are presented in the remaining chapters. The main contributions of this PhD dissertation are the following:

Chapter 1. This chapter deals with the Transit Network Timetabling and Scheduling Problem (TNTSP) in a public transit line. The TNTSP aims at determining optimal timetables for each line in a transit network by establishing departure and arrival times of each vehicle at each station. We assume that customers know departure times of line runs offered by the system. However, each user, traveling later of before their desired travel time, will give rise to an inconvenience cost, or a penalty cost if that user cannot be served according to the scheduled timetable. The provided formulation allocates each user to the best possible timetable considering capacity constraints. The problem is formulated using a p-median based approach and solved using a clustering technique. Computational results that show useful applications of this methodology are also included.

Chapter 2. The Transit Network Timetabling and Scheduling Problem (TNTSP) aims at determining optimal timetables for each line in a transit network by establishing departure and arrival times at each station allocating a vehicle to each timetable. The current models for planning timetables and vehicle schedules use the knowledge of passengers' routings from the results of a previous phase. However, the actual route a passenger will take strongly depends on the timetable, which is not yet known a priori. This chapter deals with the TNTSP in a public transit network integrating in the model the passengers' routings. The provided formulation guarantees that each user is allocated to the best possible timetable ensuring capacity constraints.

Chapter 3. Public transportation systems in metropolitan areas carry a high density of traffic daily, heterogeneously distributed, and exposed to the negative consequences derived from service disruptions. Breakdowns, accidents, strikes, require on-line operation adjustments to address these incidents in order to reduce their side effects, such as passenger extra-waiting times, complaints, potential operational dangers, etc. The Rescheduling Problem consists of defining a new schedule for a set of previously scheduled trips, given that one/several trips cannot be carried out. This chapter deals with the rescheduling problem in a transit line that has suffered a fleet size reduction. We present different modelling possibilities depending on the assumptions that need to be included in the modelization and we show that the problem can be solved rapidly by using a constrained max-cost-flow problem whose coefficient matrix we prove is totally unimodular. We test our results in a testbed of random instances outperforming previous results in the literature. An experimental study, based on a line segment of the Madrid Regional Railway network, shows that the proposed approach provides optimal reassignment decisions within computation times compatible with real-time use.

Chapter 4. In this chapter we discuss the multi-criteria p-facility median location problem on networks with positive and negative weights. We assume that the demand is located at the nodes and can be different for each criterion under consideration. The goal is to obtain the set of Pareto-optimal locations in the graph and the corresponding set of non-dominated objective values. To that end, we first characterize the linearity domains of the distance functions on the graph and compute the image of each linearity domain in the objective space. The lower envelope of a transformation of all these images then gives us the set of all non-dominated points in the objective space and its preimage corresponds to the set of all Pareto-optimal solutions on the graph. For the bicriteria 2-facility case we present a low order polynomial time algorithm. Also for the general case we propose an efficient algorithm, which is polynomial if the number of facilities and criteria is fixed.

Chapter 5. In this chapter, Ordered Weighted Average optimization problems are studied from a modeling point of view. Alternative integer programming formulations for such problems are presented and their respective domains studied and compared. In addition, their associated polyhedra are studied and some families of facets and new families of valid inequalities presented. The proposed formulations are particularized for two well-known combinatorial optimization problems, namely, shortest path and minimum cost perfect matching, and the results of computational experiments presented and analyzed. These results indicate that the new formulations reinforced with appropriate constraints can be effective for efficiently solving medium to large size instances.

Chapter 6. In this chapter, we study the multiobjective Minimum cost Spanning Tree Problem (MST) from a modeling point of view. In particular, we use the ordered median objective function as an averaging operator to aggregate the vector of objective values of feasible solutions. This leads to the Ordered Weighted Average Spanning Tree Problem (OWASTP), which we study in this chapter. To solve the problem, we propose different integer programming formulations based in the most relevant MST formulations and in a new one. We analyze several enhancements for these formulations and we test their performance over a testbed of random instances. Finally we show that an appropriate choice will allow us to solve larger instances with more objectives than those previously solved in the literature.

Related publications

The material presented in this thesis is based on the following papers, some of them published and others submitted (or about to be submitted) for publication in journals in the area of Operation Research:

Chapter 1 LOCATING OPTIMAL TIMETABLES AND VEHICLE SCHEDULES IN A TRANSIT LINE

- Mesa, J.A.; Ortega, F.A. & Pozo, M.A. (2014). "Locating optimal timetables and vehicle schedules in a transit line". *Annals of Operations Research*, (222): 439–455.

Chapter 2 INTEGRATING TIMETABLES, VEHICLE SCHEDULES AND PASSENGER ROUTINGS IN A TRANSIT NETWORK

• Laporte, G.; Ortega, F.A.; Pozo, M.A. & Puerto, J. (2015). "Optimal timetables and vehicle schedules in a transit network". *To be submitted.*

Chapter 3 On-line vehicle rescheduling in a transit line

• Mesa, J.A.; Ortega, F.A. & Pozo, M.A. (2013). "A geometric model for an effective rescheduling after reducing service in public transportation systems". *Computers & Operations Research*, vol. 40: pp. 737–746.

• Mesa, J.A.; Ortega, F.; Pozo, M.A. & Puerto, J. (2014). "Rescheduling Railway Timetables in Presence of Passenger Transfers Between Lines Within a Transportation Network".*Computer-based Modelling and Optimization in Transportation*, vol. 262: pp. 347–360.

Chapter 4 The multi-criteria p-facility median location problem on networks

• Kalcsics, J.; Nickel, S.; Pozo, M.A.; Puerto, J. & Rodríguez-Chía, A.M. (2014). "The multi-criteria p-facility median location problem on networks". *European Journal of Operational Research*, (235): 484–493.

Chapter 5 A modeling framework for Ordered Weighted Average Combinatorial Optimization

• Fernández, E.; Pozo, M.A. & Puerto, J. (2014). "A modeling framework for Ordered Weighted Average Combinatorial Optimization". *Discrete Applied Mathematics*, (169): 97–118.

Chapter 6 Ordered Weighted Average Optimization in multiobjective spanning tree problems

• Fernández, E.; Pozo, M.A. & Puerto, J. (2015). "Ordered Weighted Average Optimization in multiobjective spanning tree problems". *Submitted.*

Other contributions not included in this PhD dissertation strongly related to the research line:

1. Ortega, F.A.; Pozo, M.A. & Puerto, J. (2015). *Modelling and planning public cultural schedules for efficient use of resources* . Computers & Operations Research , (58): 9–23.

2. Mesa, J.A.; Ortega, F.A. & Pozo, M.A. (2009). "Effective Allocation of Fleet Frequencies by Reducing Intermediate Stops and Short Turning in Transit Systems". *Lecture Notes in Computer Science*, vol. 5868: pp. 293–309.

3. Laporte, G.; Mesa, J.A.; Ortega, F.A. & Pozo, M.A. (2009). "Locating a metro line in a historical city centre: application to Sevilla". *Journal of the Operational Research Society*, vol. 60: pp. 1462–1466.

Chapter 0

Preliminaries

0.1 Public transport systems

The search of consumer goods, job opportunities, studies centers, leisure and other services offered by modern society are the main motivators for the mobility of citizens. Transportation is considered a key component in the processes of production and distribution of material and cultural goods of society. In practice, no human activity can be conceived without performing displacement of people or objects, hence the dedication to solve transportation problems has been along the history a constant challenge for all models of human organization.

In the vast majority of existing urban contexts, it is commonly accepted that private modes do not provide by themselves a solution to the transportation problem in the long term, so that a significant portion of the trip demand must be satisfied by public modes. Consequently, public transport systems play a key role in the mobility of people for cities of medium and large size and they constitute the basic components for unifying the social and economic structure. The term "urban public transport" is commonly used to identify the commercial services for passengers who must pay a preset fare. A public transport system holds regular expeditions along determined routes, with known schedules and points of access. In these public transport systems, different actors can be distinguished:

- Users (passengers): People with transportation needs who are willing to spend time and money to meet it.
- Operators (Transport business): companies that provide users of transportation by means of vehicles, fuel, crew and maintenance.
- Additionally, certain agents can act as regulatory entities (local, regional and national governments), responsible for ensuring social service of transportation to city residents.

Logistics consists of planning and managing the movement and placement of goods or people, as well as the related supporting activities, all within a system designed to achieve specific objectives. When planning a transportation system, it is necessary to initially establish if the service to be provided can be considered as public or private and if the elements to be transported are people or objects. In private companies a key objective is to obtain economic benefits as a result of their activity. Therefore, transport resources will be offered only if there is a (short, medium or long term) reliable expected return. Conversely, in public companies, profit is a secondary objective and other different criteria make sense such as population coverage, the fight against social exclusion, sustainability, etc. Possible losses of profitability are assumed in the public infrastructure projects and typically the objective is to maintain an affordable level of public deficit.

0.1.1 Users' behaviour and equilibrium.

In many existing transport networks, the "non-cooperative" user behavior aggravates the problem of congestion. In general, travelers select their route from an origin to a destination in order to minimize their own travel cost (travel time). Although this choice is optimal from the perspective of the individual user, the results derived from the application of this criterion by each user can individually produce a negative effect on the community. The well-known Braess paradox illustrates this fact.

In 1968, Braess published in the journal "Unternehmensforschung" the document "Über ein aus der Paradoxon Verkehrsplanung", a reflection on the difference between the concept of optimal choice for the user and optimal situation from the perspective of the system. It seems logical that if more roads were built and the number of vehicles were increased, the traffic should be more fluid. However, in Braess's network (see Figure 1) we can see how the expansion of its configuration with adding new connection, without introducing any change in the behavior of travel demand, would cause that all travelers would incur in a higher travel cost.

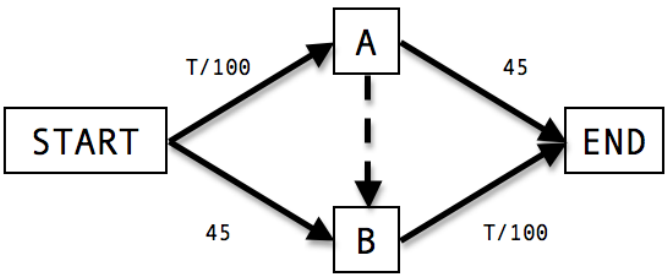


Figure 1: Braess's network

Suppose that 4000 drivers want to go from START to END in the minimum time possible. The numbers on the edges of the figure (45) indicate a fixed travel time of 45 minutes, while the labels T/100 express that the time in minutes required for traversing depends on the number of vehicles that circulate divided by 100. Assume now that there is a highway that connects the intermediate points A and B and allows a negligible travel time (0 minutes) in comparison to the total travel time. If we assume that 4000 drivers were unsupportive (i.e., they minimize their owun travel time without considering a collective vision), then they would choose the path START-A-B-END, employing a total of 80 minutes. But if 2000 drivers chose traveling through one of these two routes (START-A-END) and the other 2000 drivers took the alternative route(START-B-END), we would have a total travel time for each driver of 65 minutes (2000/100 + 45 = 20 + 45 = 65). Therefore, a collective strategy would be more beneficial to individual users. This is the reason for the existence of a central controller that makes decisions regarding the volume of flows that should be assigned to the edges of the network so as to minimize the total transportation cost.

0.1.2 Demand assignment models

Passenger assignment models aim to describe the way users of a public transportation system employ the available infrastructure for traveling between different origins and destinations in the network. Several models have been proposed, differing with respect to the assumptions on passenger behavior, network structure, and modeling of congestion (Cominetti and Correa, 2001).

One important concept in demand assignment is the determination of the minimum cost path. Important elements in defining the attributes of cost include:

1. The characterization of time-dependence and stochastic attributes in the minimum cost path.
2. The characterization of a solution as: (i) a single path, including only a route or combination of routes; (ii) a path that can include a set of common lines (Cominetti and Correa, 1999), including cases where multiple routes may overlap on some part of the shortest path; or (iii) a strategy, allowing passengers to choose their own boarding rules as they travel from origin to destination (Spiess and Florian, 1989).
3. The effect of capacity and congestion in the transit network.

Demand behaviour has been deeply studied in passenger assignment models in order to describe the way users of a public transportation system employ the available infrastructure for traveling between different origins and destinations (Pel et al., 2014). Two families of assignments models are usually used to predict user's behaviour inside a transportation network: frequency-based and schedule-based models.

Frequency-based models (FBM) are based in the statement that passengers choose their strategies according to the level of frequencies established in the lines of the network since frequencies determine waiting times and load in vehicles (Cepeda et al., 2006). Two main drawbacks have been shown in this approach. First, the behaviour of passengers vary according to waiting times. Small frequencies lead users to arrive randomly to stations but when frequencies are bigger, users tend to follow strictly timetables, arriving only few minutes before the departure. In this sense, frequency-based models do not allow computing *user's schedule costs* (Small, 1982) that is, deviations from desired and actual travel time, and allocations to lines are made by distributions or probabilities. The second drawback comes from the fact that FBM are not sufficient to face the dynamic nature of the capacity problem, so it is often required to deal with fail-to-board probabilities (Schmöcker et al., 2008) that only allow obtaining average line loadings over the modeling period, instead of line loadings for specific services. The main advantage of frequency-based approaches is that less detailed input data are required.

In the last 20 years, *schedule-based models* have been used for detailed operational planning of services with low frequencies (Nuzzolo et al., 2012; Hamdouch et al., 2011, 2014). The computational demand of schedule-based models has often been a major obstacle; however, recent computational improvements also made it possible to use schedule-based models for larger networks and/or for networks with a high density of services, like metropolitan subway networks (Poon et al., 2004). The inherent advantage of schedule-based over frequency-based models is that schedule-based approaches always consider dynamic effects and allow tracking the time at which passengers pass each node on their way from their origin to their destination.

0.2 The global transportation planning process

A public transportation design involve decisions in three different scopes:

1. Demand forecasting. Implies the study and updating of present and future user's flows between different points of a city.

2. Mode choice. That is, selection of the mean of transport that will operate between different locations. This decision, should take into account, speed, capacities, constraints related to geographical, environmental and economical restrictions as well as compatibilities with other means of transport already in use.

3. Transportation planning. Includes routes design, stations and depots location and transport policies regarded with the level of service determined.

Ceder and Wilson (1986) establish five steps in the global transportation planning process: line planning, frequencies setting, timetabling, vehicle scheduling and crew rostering. Some transportation systems require first a traditional network design problem consisting of selecting edges (rails, roads, etc) and nodes (stations, depots, parkings, etc). Over this infrastructure, lines and routes are defined. Figure (2) shows the whole planning process as a systematic decision sequence. It is obvious that the order and independence of this activities exist only in the diagram, since any decision made in upper leves will have consequences in lower levels.

Some authors, treat this process as sequence of three levels of decision (Van de Velde, 1999):

1. The strategic level, that comprises long-term decisions (5 years approximately) such as decisions in network design and line planning.

2. The tactical level, that comprises decisions valid in medium-term (1-2 years) such as the frequencies setting and the timetabling.

3. The operational level, that implies short-term decisions, taken once a day/month, like the vehicle scheduling and crew rostering.

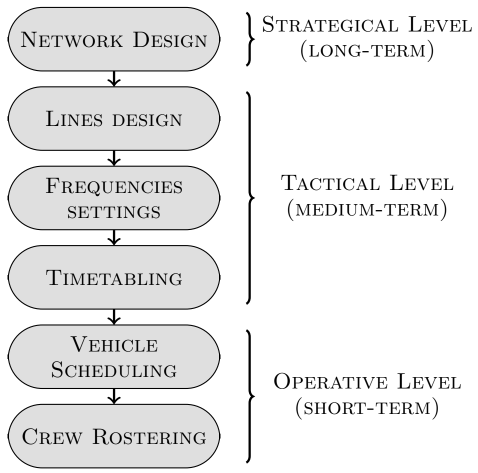


Figure 2: Global planning process

The main goal of most transit operators is to offer to the population a service of good quality that allows passengers to travel easily at a low fare. The operators thus have a social mission which aims at reducing pollution and traffic congestion, as well as increasing the mobility of the population (Desaulniers and Hickman, 2007). In most cases, the goal is usually not to make profits, as is the case for almost all other transportation organizations such as airlines, railroads, and trucking companies. They are, however, subject to budgetary restrictions that force them to manage expensive resources such as buses, drivers, maintenance facilities and bus depots as efficiently as possible. Briefly stated, the global problem faced by the agencies consists of determining how to offer a good-quality service to the passengers while maintaining reasonable asset and operating costs.

0.2.1 Strategic Level

Transit network design

Given a potential network, the Transit Network Design Problem (TNDP) consists basically of selecting a subset of nodes (access and exit nodes to network) and edges (connections) that will allow a potential demand traveling between the nodes of the network. The accessibility to the network increases with the number of stations but, on the other hand, the commercial speed between an origin and a destination decreases according to the number of intermediate stops (Murray, 2003; Schöbel, 2005). The biggest constraint over the number of edges selected comes from the kind of network in study (bus, tram, subway) and the type of construction (surface, tunnel, raised platforms). The line planning step with/without the frequencies setting can be included in the TNDP or they can be left to be solved in as subsequent problems.

From the user's perspective, a transit network should cover a large service area, being easily accessible and offering a large number of possible trips in order to meet globally the demand. On the other hand, the number of stations and the network size should be restricted in order to control operator's costs.

Building a transit network can be an expensive and complex task where a widespread variety of decision-makers have influence; such as urban planners, traffic and civil engineers, politicians, geologists, environmentalists and citizen interest groups. None of the available methods can claim to provide a complete and definite solution to an outstanding planning problem (metro, rail, roads) as there are just too many players, criteria and uncertainties present in such projects. However, operational research can be useful in generating and assessing alternative solutions and in solving specific subproblems. As an example, the construction of a metro link under the historical centre of a city leads to a multi-criteria optimization problem where alignments should be far from fragile buildings, to reduce the effects of vibrations, without deviating too much from the most direct trajectory (Laporte et al., 2009).

0.2.2 Tactic level

Line planning and frequency setting

The *Transit Network Frequencies Setting Problem (TNFSP)*, consists of determining a set of lines and associated frequencies for a given infrastructure network and a demand pattern. Objectives and constraints are usually related with the directness of the routes, the service coverage and the operational costs.

Yu et al. (2005) claim that the design of lines and the frequencies setting should not be treated simultaneously, since the line network is a component more stable than a flexible parameter as the frequency. However, it is of special interest to relate both problems in order to get better results, especially when demand is assumed to be endogenous and determined as an equilibrium problem, in which the flows are a function of the network design (see e.g., Gao et al., 2004). With the origin-destination flows and an assignment of these flows to routes, the set of routes and their frequencies must be determined.

From the users' perspective, the layout of the lines should cover a large service area, offering a big number of direct trips close as possible to shortest paths and it should meet the global demand. Complementary, the number of stations and the total length of the network should stay below an upper bound in order to reduce operator's costs. In addition, an appropriate frequencies assignment for each line should be adequate for each time period of the day, week or season of the year. Furthermore, the service should be regular enough in order to satisfy the users' demand but sparse enough so as to limit the required fleet size and therefore, the operational costs.

Timetable design

The *Transit Network Timetabling Problem (TNTP)* consists of establishing a timetable for each (predetermined) line of the transit network by means of departure and arrival times at each station. This process can be dependant on a preestablished set of frequencies, or it can be independent to determine jointly frequencies and timetables. In addition, the policy of headways could be given implying a minimum/maximum frequency imposed as constraints in the timetables.

With the aim of reducing waiting times at interchange zones, transfers can be coordinated especially when headways are long (Guihaire and Hao, 2008). In this case, losing a connection may incur in big delays but the lack of synchronization may discourage users, avoiding such transport system if possible. On the other hand, high frequencies incur just waiting few minutes in case of losing a connection (Chakroborty et al., 2001) and, in case of congestion, a transfer coordination may lead to unreliable transfers.

There are two main timetable variants. One variant is the periodic (or cyclic) timetable that is repeated every given time period, for example every hour, with only slight differences between peak hours and off-peak hours. The other variant is the non-periodic timetable, that allows to follow the passenger demand with the frequencies of the trains. In both cases the timetable is usually repeated every day, although there may be differences between weekdays and weekend days. Periodic timetables are suitable for rapid transit systems, where the frequency of the vehicles is high and their departures are equally spaced. In the case of low frequency transportation systems, periodic timetables are easy for the passengers to remember. Assuming constant demand, periodic timetables provide minimum waiting times (Larson and Odoni, 1981). However, a periodic timetable applied to a general demand case leads to unbalanced levels of occupancy of the trains and higher waiting times. Conversely, demand behaviour also depends on the level of frequencies/timetables. Narrow headways lead users to arrive randomly to stations giving rise to a waiting cost for the user. On the other hand, when headways are wider, customers tend to strictly follow timetables, arriving only few minutes before the departure time. This last situation does not provoke a waiting cost for the user but an inconvenience cost to fit desired travel time to actual timetables (Grosfeld-Nir and Bookbinder, 1995; Fosgerau, 2009). The concept of schedule delay (Small, 1982) arises with the fact that arriving early is likely to involve some wasted time while for most users, arriving late has more severe repercussions. In this way, timetabling can be seen as a p-median problem (Hakimi, 1964) where the objective is to minimize the time/distance between passenger desired departure times and actual ones (Mesa et al., 2013).

Demand satisfaction in the TNTP can be measured in terms of travel time (since timetable setting permits to compute passengers travel time for the first time in the process), transfers (in both terms of possibilities and synchronization) and schedule delays. From the operator's point of view, a proper timetable setting may help to reduce the required fleet size.

0.2.3 Operational level

Operational level I: Vehicle scheduling

The *Vehicle Scheduling Problem* (VSP) consists of allocating vehicles to a given set of timetables, considering some practical requirements such as multiple depots, types of vehicles and other extensions. An optimal vehicle schedule minimizes the fleet size as well as the operational costs. More precisely, given a set of travel times for predetermined trips (with established departure and arrival times) as well as starting and finishing locations, the VSP is to find the vehicle allocation to trips such that: (1) each trip is performed once, (2) each vehicle performs a feasible sequence of trips and (3) the total costs are minimized. The total costs can be divided in fixed costs of vehicles (as investments and maintenance) and operational costs (as fuel and wear). Operational costs can be also understood in different ways taking into account travel distances, travel times or even waiting times. The VSP has been widely studied in the last 40-50 years (see, e.g., Törnquist, 2007; Bunte and Kliewer, 2009).

Operational level II: Crew rostering

The crew rostering problem consists of allocating a set of drivers to a set timetabled trips in order to minimize the cost of duties and satisfy labor regulations constraints. The Driver Scheduling Problem defines the generic daily duties whereas the Driver Rostering Problem determines the assignment of drivers to the daily duties yielded by the Driver Scheduling Problem solution for a specific planning period, e.g. a month. This assignment, called roster, must comply with labor rules and the company's regulations (e.g. maximum working days in a row). Further information on crew scheduling and rostering is analyzed in Wren and Rousseau (1995).

0.2.4 Strategies for managing interruptions and the rescheduling problem

Railway systems in metropolitan areas carry a high density of traffic daily, heterogeneously distributed, and exposed to the negative consequences derived from service disruptions. Some examples of possible disruptions are: (1) interruptions coming from severe weather conditions, accidents, and the blockage of road or tracks sections or (2) fleet size reductions coming from vehicle breakdowns, drivers' and crew strikes or vehicle reallocations made to reinforce other sections of the transit network. In particular, a scheduled timetable may become infeasible simply due to a heavy passenger flow (Mesa et al., 2009). To address these incidents, operation adjustments are required in order to reduce the side effects of emergency incidents, such as passenger waiting/traveling times, complaints, potential operational dangers, etc. The rescheduling problem consists of defining a new schedule for a set of previously scheduled trips, given that one or several trips have been severely disrupted. While many objectives and constraints remain from the timetabling problem, new requirements and objectives arise in this context. In terms of transportation of people, the main decisions concern to the minimization of the deviations from the initial timetable in operation and delay costs.

Rescheduling is the process of updating an existing production plan in response to disturbances or disruptions (Vieira et al., 2003). Customers plan their trips based on a known timetable, and can be greatly inconvenienced if the service does not arrive or depart at the expected time. When a disturbance occurs, like a train breakdown in a certain line, the system operator must make a decision about rescheduling the remainder vehicles which are normally operating along the network in order to reduce the loss of service quality perceived by the users. An important difference between the planning stage and the rescheduling stage during disruptions is that in the latter case less time is available for rescheduling. In principle, solutions are expected within minutes (on-line). For the resources, another important difference is that in general there is less flexibility in the rescheduling stage, since many resource duties have already started at the time of the disruption when the rescheduling is carried out, and cannot be easily diverted. In addition, the solution space is bounded by the remaining time until the end of the rescheduling horizon, which is usually the end of the day. Hence, if the disruption happens in the evening, then the solution space is much smaller than in case the disruption happens in the morning. For example, a straight forward *myopic* strategy consists in canceling those services that serve to the least number of users. This methodology would not introduce any change/delay in the remaining timetables. Nevertheless, a seminal paper by Mesa et al. (2013) has shown that if real time control strategies are applied along a transit corridor (i.e., by allowing delays at some services of the initial schedules), then the demand satisfaction after rescheduling can be increased significantly.

Altay and Green (2006) suggest that emergency response efforts consist of two stages; pre-event and post-event response. Pre-event tasks include predicting and analyzing potential dangers and developing necessary action plans for mitigation. Post-event response starts while the disaster is still in progress. At this stage the challenge is locating, allocating, coordinating, and managing available resources. In these terms, we focus next on robust design and rescheduling operations.

The adequateness of a transport system to accommodate itself to perturbations or disruptions is defined as the robustness of a transit network. Often, changes in the initial plan take place within a transit network, giving rise to incidences ranging from small deviations to long interruptions. However, transit systems for the optimization of itineraries and frequencies do not usually consider the possible appearance of changes in the demand pattern or failures in the system performance. Therefore, even if correctly designed, a transit network can collapse after an incidence in a node or arc (station or road/tracks) leaving unfulfilled the objectives for which the network was initially designed (Wirasinghe, 2003). Therefore, the analysis of the infrastructures vulnerability and their robust design are areas of strong interest.

A first approach to the study the vulnerability of a network consists of studding its robustness against random failures (Altay and Green, 2006) such as breakdowns (failures of vehicles, signaling, blockage of network edges) human actions (accidents, strikes) or natural disasters (flooding, earthquakes, etc). Another point of view for disruption management consists of its study against attacks and worst case scenarios (Matisziw and Murray, 2009) in the network infrastructure (e.g. as a consequence of strikes, sabotages or terrorist attacks). Many network design techniques stress their interest in including some kind of redundance in the network topology providing extra connectivity to keep the performance against failures (Lozano et al., 2008). Besides, the design of a new alignment can keep some construction-safety criteria (Laporte et al., 2009) or it can be done providing an extra connectivity to the network in a worst case scenario (Mesa et al., 2008).

Delays and interruptions management (Liebchen et al., 2010) is an important task in the operations control of every public transport company. The partial or complete cancellation of services implies a special care in the decision making to decide how to reschedule the existing resources in order to offer a service as similar as possible to the one before the incidence (Mesa et al., 2013). In addition, a set of external vehicles can be coordinated in order to face emergencies (Arriola et al., 2009) or a heavy passenger flow for example at peak hours, massive events, etc (Mesa et al., 2009).

The common strategies for managing disturbances and disruptions are the following:

- Reduction of travel times between stations. Timetables often include buffer times along a line run that can be reduced as needed.
- Deadheading. It is a control strategy that consists of skipping a number of stops (usually with low demand) at the beginning of the line when the vehicle is still empty.
- Express service. Skips a set of intermediate stations along a transit line. Passengers must be informed about which destinations will not be reachable.
- Short turning. Consists in changing the direction of a vehicle before reaching a terminal station shortening the cycle performed by the vehicle along the transit line. In this case, passengers must be informed about which station will be the last destination.
- Coordination with other alternative means of transport.
- Delay of some services. If the fleet supply after a disruption is not enough to cover all trips, one possibility is to delay some services in order to serve more demand. This strategy should take into consideration the capacity level of vehicles and connections with other lines or means of transport.
- Cancelation of services. When the presence of disturbances make not possible to keep the same level of service as before the disruption, it becames crucial to decide which services have to be cancelled.

The development of algorithmic real-time railway rescheduling methods is currently still mainly an academic field, where the research is still far ahead of what has been implemented in practice. Unfortunately, the public transport industry has never been a quick adopter of newly available and innovative methods and concepts. Nevertheless, there are signals that show the interest for the added value that can be provided by real-time rescheduling methods, based on the successes that have been achieved by the application of optimization methods in the railway planning stage.

0.3 Location theory

A location problem consists of determining the position of one or more facilities in order to optimize a measure of effectiveness with respect to a set of known demand locations. Location theory, as any other discipline in Operations Research, develops mathematical models to reflect, as good as possible, the real situation being studied, with adequate solutions to the problem under study. This area of research has already a long history and it is now in full expansion since a lot of methods and procedures can successfully be exported in order to solve complex problems belonging to other knowledge areas.

Location problems can be classified into three categories: discrete location, location on networks and continuos location. Discrete location imposes that the set of candidate locations for placing the new facility(ies) is finite. Network location problems assume demand points within a graph and facilities have to be located in the nodes or edges of the graph. Finally, continuos location considers problems where demand points are within a continuos space, typically an Euclidean space. This section is mainly focused on discrete location and location on networks problems, that will appear in the subsequent chapters. Excellent references that cover these fields in Location Theory are Larson and Odoni (1981); Mirchandani and Francis (1990); Daskin (1995, 2013) and other references covering all fields are Drezner (1995); Drezner and Hamacher (2002); Puerto (1996); Laporte et al. (2015).

In order to get a better understanding of the location problems structure, we briefly describe next the common elements to all of them.

The solution space:

The solution space is the framework where the problem is defined. It contains as elements existing facilities and the new facility(ies). The choice of an appropriate solution space is crucial, because it determines aspects as important as the accuracy and solvability of the model. Some usual solution spaces are:

- *Discrete spaces:* When there exists a finite number of potential locations for the new facilities.
- *Networks:* The solution candidates lie within a graph, usually representing a communication network. Nodes represent important elements, such as cities or crossroads. Arcs represent connections between nodes, like roads, streets, cables, etc. A kind of network that has received considerable attention is the "tree network". This is due mainly to the uniqueness of a path between pairs of points.
- *Euclidean space $\mathbb{R}^n$:* It is used when the problem presents regional aspects that cannot be discretized. In addition, it can be used to approximate networks when the number of nodes and arcs is quite big.
  The cases $n = 2$ and $n = 3$ have a clear physical meaning. Cases where $n \geq 4$ have been used to model and solve estimation problems in statistics.
- *Sphere:* It is useful for those real situations that cope with large scale distances.

- *Embedded network in a continuos space:* This is the solution space where a network, that represents high speed connections, overlaps an Euclidean space or a sphere.

Existant locations:

In terms of Location Theory, *existing facilities* are the users that require to be served. Therefore, they are called demand points. Usually, they are modelled by means of a set D and an *intensity* function to weight the elements of D.

There exist two different ways of representing demand in the solution space: by a finite set of points and by regions. In the first case, a set of points $D = \{d_1, \ldots, d_M\}$ is considered as well as a set of *weights* $\{w_1, \ldots, w_M\}$ that represent the importance (or intensity) of the demand generated at each point. In the regional model, demand is represented by means of a region $\mathcal{R}$ (not necessarily connected) included in the solution space and it is a probability measure which gives importance to each measurable subset of $\mathcal{R}$.

The new facility(ies):

The location of the new facility is the decision variable of the general location problem. This variable is characterized by

a) Number and quality of the service provided. If more than one facility is to be located, it will be necessary to specify the characteristics of each one of them. When they are identical, as for instance mail boxes, we face with a multifacility problem; otherwise as in the case of health services, we can find hierarchical location problems.

b) Nature of the service. Not all the services are attractive for the community where they will be located. For instance, nuclear plants, solid waste disposals or garbage plants are usually refused by population. Therefore, in modelling a problem it is very important to determine the attractiveness of the service.

The objective function:

Location problems mentioned in this section have the following objective function in common:

$$\operatorname*{opt}_{X=(x_1,\ldots,x_p)\subset S} F\Big(d(X,a)_{a\in D}\Big)$$

where

F is a globalizing function,

"opt" means optimize, either minimize or maximize,

S is the solution space,

$X = \{x_1, \ldots, x_p\} \subset S$ is the new facility(ies), either single $p = 1$ or multiple $p > 1$.

D is the set of existing facilities (demand points),

a is a general existing facility

$d(\cdot, \cdot)$ is a mesure of distances. In general, $d(X, a)$ stands for the distance between demand a and the set of facilities $(x_1, \ldots, x_d)$.

The determination of which objective function has to be used is sometimes a hard task. It should be noted that the final solution strongly depends on that choice. Therefore, it is important to devote some effort to this part of the modelling process.

1. *The p-median problem or "minisum".* The p-median problem (Hakimi, 1964, 1965) searches for the location of p facilities with the opjective of minimizing the weighted sum of distances between the demand points and the facilities to which they are located. A general p-median formulation is the following:

$$\min_{X \subset S} \sum_{a \in D} d(X, a)$$

   The contribution of mean distance models can be interpreted as the search of an efficient objective for example when an economical criterion is imposed.

2. *The p-center problem problem or minmax.* The p-center problem assumes that all the demand is covered with p facilities and minimizes the coverage distance for doing so, that is, the maximum weighted distance between a demand and its nearest facility is minimized as follows:

$$\min_{X \subset S} \max_{a \in D} d(X, a)$$

   The minmax model can be interpreted as a quality criterion of the developed service in terms of equity.

3. *Centdian problem.* Given a positive scalar $\lambda \in (0, 1)$, the objective function corresponds now to a convex combination of the criteria minisum and minmax. That is, the problem is:

$$\min_{X \subset S} (\lambda \sum_{a \in D} d(X, a) + (1 - \lambda) \max_{a \in D} d(X, a))$$

   The cent-dian model corresponds to a compromise between the center and median criteria, that are conflicting criteria in most of the cases.

4. *Ordered median problem.*

Given a finite number of existing facilities $D = \{a_1, \ldots, a_M\}$ and weights $\lambda_1, \ldots, \lambda_M$, the objective is to find the location of X minimizing an ordered sum of distances, i.e.,

$$\min_{X \subset S} \sum_{i=1}^{p} \lambda_i d_{(i)}(X).$$

Here, $d_{(i)}(X) = d(X, a_{\sigma_i})$ is the i-th element in the list of sorted distances

$$d(X, a_{\sigma_1}) \leq \ldots \leq d(X, a_{\sigma_M})$$

where σ is a permutation of $\{1, ..., M\}$. Note that this objective function is point-wise defined, because it expression changes when the order between distances is modified. This function is somehow similar to the p-median, but is more general because it includes as particular instances the minsum, minmax and centdian (among others not referenced in this list).

5. *Set covering problem.* In this problem, the number of facilities to be located is not fixed *a priori*, that is, the cardinality of X (denoted by $|X|$) has to be minimized and determined together with its elements. Each existing facility s should be within a specified distance from at least a new facility r_a. The objective is to find the lowest number of facilities and their location verifying the above constraint. Thus, the problem can be written as:

$$\min_{X \subset S : d(X,a) \leq r_a, a \in D} |X|.$$

6. *Maximal covering problem.* The objective of this problem is to have as many existing facilities within a specified distance, called the covering distance, from a nearest facility. Unlike the set covering model, the set of new facilities is fixed to p. Let us consider $\delta(d(X, a)) = 1$ iif $d(X, a) \leq r_s$; 0 otherwise. The problem is:

$$\max_{X \subset S} \sum_{a \in D} \delta(d(X, a))$$

7. *Multiobjective problem.*

The previous objectives establish *a priori* the criteria used to locate the new facility(ies). However, there exist real situations where it would be reasonable to use simultaneously several criteria. This imply to find solutions that are optimal to several criteria at the same time. This type of problems are called multiobjective location problems. Given that the different criteria are usually in conflict, the "ideal" solution rarely exists and, therefore, one has to decide which concept of "optimality" to choose. For example, the non-dominated solutions are those that cannot be improved for all objectives by any other solution.

For a detailed discussion on the nature of multiobjective location problems we refer to Nickel et al. (2005) and Nickel et al. (2015). We provide more insight in multiobjective optimization in the following section.

0.4 Multiobjective optimization

Many real-world applications are concerned with finding an optimal location for one or more new facilities minimizing a function of the distances between these facilities and a given set of existing facilities (clients, demand points). Most of the existing research focuses on the minimization of a single objective function that is increasing with distance. However, in the process of locating a new facility usually more than one decision maker is involved. This is due to the fact that often the cost incurred with the decision is relatively high. Furthermore, different decision makers may (or will) have different (conflicting) objectives. Therefore it will be very probable that a locational decision is made by a group of q decision makers (DM). Even if each DM chooses the same objective function to evaluate the quality of a new location, the weights assigned to clients may differ a lot. In other situations, different scenarios must be compared due to uncertainty of data or still undecided parameters of the model. One way to deal with these situations is to apply scenario analysis. Another way of reflecting uncertainty in the parameters is to consider different replications of the objective function. Hence, there exists a large number of real-world problems which can only be modelled suitably through a multicriteria approach, especially when locating public facilities.

Multicriteria analysis of location problems has received considerable attention within the scope of continuous, network, and discrete models in the last years. For an overview of general methods as well as for a more bibliographic overview of the related location literature the reader is referred to Ehrgott (2005), Nickel et al. (2005) and Nickel et al. (2015). Presently, there are several problems that are accepted as classical ones: the point-objective problem (see, e.g., Wendell and Hurter, 1973; Hansen, 1980; Carrizosa et al., 1993), the continuous multicriteria min-sum facility location problem (see, e.g., Hamacher and Nickel, 1996; Puerto and Fernández, 1999), the network multicriteria median location problem (see, for instance, Hamacher et al., 1999) and the multicriteria discrete location problem (see, e.g., Fernández and Puerto, 2003), among others.

The goal in a multicriteria location problem is to optimize simultaneously a family of objective functions $F(X) = (F^1(X), \dots, F^k(X))$. Therefore, the formulation of the problem is:

$$\operatorname*{v-min}_{X \in S} F(X)$$

where v-min stands for vector minimization, X is the new facility to be located and S is the solution space.

Observe that we get points in a k-dimensional objective space so, in contrast to problems with only one objective, we do not have a natural ordering in higher dimensional objective spaces. Therefore, in multicriteria optimization one has to decide which concept of "optimality" to choose.

Accordingly, for this type of problems, different concepts of solution have been proposed in the literature We assume the usual definition of Pareto-optimality or efficiency (Ehrgott, 2005). That is, a solution X is called efficient or Pareto-optimal, if there exists no solution X' which is at least as good as X with respect to all objective function values and strictly better for at least one value, i.e.,

$\nexists X' : F^q(X') \leq F^q(X), \forall\, q \in Q = \{1, ..., k\}$, and $\exists q \in Q : F^q(X') < F^q(X)$. If X is Pareto-optimal, $F(X) \in \mathbb{R}^k$ will be called a non-dominated point. We denote by $\tilde{X}$ the set of all Pareto-optimal solutions. If $F^q(X) \leq F^q(X')\ \forall\, q \in Q$ and $\exists q \in Q : F^q(X) < F^q(X')$ we say X dominates X' in the decision space and $F(X)$ dominates $F(X')$ in the objective space.

Alternative solution concepts are weak Pareto-optimality and strict Pareto-optimality. A solution X is called weak Pareto location (or weakly Pareto-optimal), if there exists no solution X' which is at least as good as X with respect to all objective functions, i.e., $\nexists X' : F^q(X') < F^q(X), \forall\, q \in Q$. We denote by $\tilde{X}_w$ the set of all weak Pareto-optimal solutions. A solution X is called strict Pareto location (or strictly Pareto-optimal), if there exists no solution $X' \neq X$ at least as good as X with respect to all objective functions, i.e., $\nexists X' : F^q(X') \leq F^q(X), \forall\, q \in Q$. We denote by $\tilde{X}_s$ the set of all strict Pareto-optimal solutions. Note that $\tilde{X}_s \subseteq \tilde{X} \subseteq \tilde{X}_w$ and in case we are considering strictly convex functions these three sets coincide. Finally, we recall that Warburton (1983) proved the connectedness of the set $\tilde{X}$ when the functions are convex.

0.4.1 Discrete Location Problems

Planar and network multicriteria location problems have been widely developed from a methodological point of view so that important structural results and algorithms are known to determine solution sets (Nickel and Puerto, 2005). On the contrary, multicriteria analysis of discrete location problems has attracted less attention. In spite of that, several authors have dealt with problems and applications of multicriteria decision analysis in this field. An annotated bibliography with many references up to 2005 can be found in Nickel et al. (2005). In general, very few papers focus in the complete determination of the whole set of Pareto-optimal solutions. Nevertheless, there are some exceptions, such as the paper by Ross and Soland (1980) that gives a theoretical characterization but does not exploit its algorithmic possibilities, as well as the work by Fernández and Puerto (2003) that addresses the computation of the entire set of Pareto-optimal solutions of the multiobjective uncapacitated plant location problem.

Nowadays, Multi-Objective Combinatorial Optimization (MOCO) (see Ehrgott and Gandibleux 2000; Ulungu and Teghem 1994) provides an adequate framework to tackle various types of discrete multicriteria problems such as, for instance, the p-Median Problem (p-MP). Within this emergent research area, several methods are known to handle different problems. It is worth noting that most of MOCO problems are NP-hard and intractable (see Ehrgott and Gandibleux 2000, for further details). Even in most of the cases where the single objective problem is polynomially solvable the multiobjective version becomes NP-hard. This is the case of spanning tree problems and min-cost flow problems, among others. In the case of the p-MP, the single objective version is already NP-hard. This ensures that the multiobjective formulation is not solvable in polynomial time unless P=NP. In this context, when time and efficiency become a real issue, different alternatives can be used to approximate the Pareto-optimal set. One of them is the use of general-purpose MOCO heuristics (Gandibleux et al. 2000). Another possibility is the design of "ad hoc" methods based on one of the following strategies: 1) computing supported non-dominated solutions; and 2) performing partial enumerations of the solutions space. Obviously, the second strategy does not guarantee the non-dominated character of all the generated solutions although the reduction in computation time can be remarkable.

Model and Notation Let $I = \{1, \dots, M\}$ and $J = \{1, \dots, N\}$ respectively denote the sets of indices for demand points and for plants, and $Q = \{1, \dots, k\}$ denote the set of indices for the considered criteria. For each criterion $q \in Q$, let $(c_{ij}^q)_{i\in I, j\in J}$ be the allocation costs of demand points to plants. The multicriteria p-median location problem is:

$$\text{v-min} \left(\sum_{i\in I} \sum_{j\in J} c_{ij}^1 x_{ij}, \dots, \sum_{i\in I} \sum_{j\in J} c_{ij}^k x_{ij} \right) \quad (1)$$

$$\text{subject to} \quad \sum_{j\in J} x_{ij} = 1, \quad i \in I, \quad (2)$$

$$x_{ij} \le y_j, \quad i \in I,\ j \in J, \quad (3)$$

$$\sum_{j\in J} y_j = p, \quad (4)$$

$$x_{ij} \in \{0,1\},\ y_j \in \{0,1\}, \quad i \in I,\ j \in J. \quad (5)$$

As it is usual, v-min stands for vector minimum of the considered objective functions. Here variable y_j takes the value 1 if plant j is open and 0 otherwise. The binary variable x_{ij} is 1 if the demand point i is assigned to plant j and 0 otherwise. Constraints (2), together with integrality conditions on the x variables, ensure that each demand point is assigned to exactly one plant, while constraints (3) guarantee that no demand point is assigned to a non-open plant. Finally, constraint (4) ensures that exactly p plants are opened.

Recall that in the single criterion case the integrality conditions on the x variables need not be explicitly stated. The reason is that when x_{ij} represents the proportion of demand of client i satisfied by plant j (i.e. $0 \le x_{ij} \le 1$), there exists an optimal solution with $x_{ij} = 0, 1$, $i \in I$, $j \in J$. This property is not necessarily true when multiple criteria are considered because, in general, there might be undominated solutions with non-integer values and even non-supported undominated integer solutions.

0.4.2 Network Location Problems

Problem definition

Let $G = (V, E)$ be an undirected connected graph with node set $V = \{v_1, \dots, v_n\}$ and edge set $E = \{e_1, \dots, e_m\}$. Each edge $e \in E$ has a positive length $\ell(e)$, and is assumed to be rectifiable. Let $A(G)$ denote the continuum set of points on edges of G. We denote a point $x \in e = [u, v]$ as a pair $x = (e, t)$, where t $(0 \le t \le 1)$ gives the relative distance of x from node u along edge e. For the sake of readability, we identify $A(G)$ with G and $A(e)$ with e for $e \in E$. Let $k \ge 1$ be the number of criteria of the problem and define $Q = \{1, \dots, k\}$. Each vertex $v_i \in V$ has a real-valued weight $w_i^q \in \mathbb{R}$, $q \in Q$. Let $J = \{1, \dots, p\}$, where p is the number of facilities to be located. We denote by

$X = (x_1, \ldots, x_p)$ the vector of locations of the facilities, where $x_j \in G$, $j \in J$. (Note that in order to allow co-location, which is quite common in location problems with negative weights, we have to represent the facility locations using a vector.) In the remainder, we use the notions location vector and solution synonymously.

We denote by $d(x, y)$ the length of the shortest path connecting two points $x, y \in G$. Let $v_i \in V$ and $x = ([v_r, v_s], t) \in G$. The distance from v_i to x entering the edge $[v_r, v_s]$ through v_r (v_s) is given as $D_i^+(x) = d(v_r, x) + d(v_r, v_i)$ ($D_i^-(x) = d(v_s, x) + d(v_s, v_i)$). Hence, the length of a shortest path from v_i to x is given by $D_i(x) = \min\{D_i^+(x), D_i^-(x)\}$. As $d(v_r, x) = t \cdot \ell(e)$ and $d(v_s, x) = (1 - t) \cdot \ell(e)$, the functions $D_i^+(x)$ and $D_i^-(x)$ are linear in x and $D_i(x)$ is piecewise linear and concave in x, cf. Drezner (1995). The distance from v_i to its closest facility is finally defined as $D_i(X) = \min_{j \in J} D_i(x_j) = \min_{j \in J}\{D_i^+(x_j), D_i^-(x_j)\}$. In the following, we call the functions $D_i^{+/-}(x)$ and $D_i(X)$ distance functions of node v_i. Moreover, we say that $D_i^a(x_j)$, $a \in \{+, -\}$, is active for X, if $D_i^a(x_j) = D_i(X)$.

We consider the objective function $F(X) = (F^1(X), \ldots, F^k(X))$, where each $F^q(X)$, $q \in Q$, is a median function defined as:

$$F^q(X) = \sum_{i \in V} w_i^q \, D_i(X) \, .$$

The k-criteria p-facility median location problem on networks, denoted by (k, p)-MLPN, is now defined as the problem of determining the set of all Pareto-optimal solutions on the graph:

$$\operatorname*{v-min}_{X \in G \times \overset{p}{\ldots} \times G} F(X) \, , \tag{6}$$

where v-min stands for vector minimization. We denote by $\tilde{X}$ the set of all Pareto-optimal solutions. As mentioned in the introduction, we are interested in obtaining a description of the complete sets of Pareto-optimal solutions (in the decision space) and the non-dominated points (in the objective space). Hereby, the set of Pareto-optimal solutions comprises all alternative location vectors for the p facilities that are suitable candidates to choose from, because no other point can give rise to objective values that dominate them component-wise.

Let $h = (e_{h_1}, \ldots, e_{h_p})$ be a p-tuple of not necessarily distinct edges, where $e_{h_j} \in E$, $j \in J$. Then, the (k, p)-MLPN can be equivalently formulated as:

$$\text{v-min}\{F(X) \mid X \in e_{h_1} \times \ldots \times e_{h_p}, \, h \in E \times \ldots \times E\} \, .$$

Note that because of symmetry it is sufficient to consider only p-tuples h for which $h_1 \leq \cdots \leq h_p$.

Solution approaches so far

Concerning the methodological aspects of multicriteria network location problems, Hamacher et al. (1999) discuss the network 1-facility problem with median objective functions. They show that for

Pareto-optimal locations on undirected networks no node dominance result can be proven. Hamacher et al. (2002) provide a polynomial time algorithm for the 1-facility problem when the objectives are both weighted median and anti-median functions. The method is generalized for any piecewise linear objective function. Zhang and Melachrinoudis (2001) develop a polynomial algorithm for the 2-criteria 1-facility network location problem maximizing the minimum weighted distance from the service facility to the nodes (maximin) and maximizing the sum of weighted distances between the service facility and the nodes (maxisum). Skriver et al. (2004) introduce two sum objectives and criteria dependent edge lengths for the 1-facility 2-criteria problem. Nickel and Puerto (2005) solve the 1-facility problem when all objective functions are ordered medians. Colebrook and Sicilia Colebrook and Sicilia (2007a,b) provide polynomial algorithms for solving the cent-dian 1-facility location problem on networks with criteria dependent edge lengths and facilities being attractive or obnoxious.

Despite its intrinsic interest as discussed above, to the best of our knowledge there are no papers discussing the multicriteria p-facility median location problem on networks and no results are known until the moment to obtain the set of Pareto-optimal solutions.

Other Multicriteria Location Problems on Networks In the recent survey Nickel et al. (2015) an overview on other location problems can be found. In Hamacher et al. (2002) an extension to 1-facility center problems as well as to positive and negative weight vectors on the nodes is developed. Those ideas have been further extended to problems with criteria dependent lengths in Skriver et al. (2004). A unified framework for multicriteria ordered median functions can be found in Nickel and Puerto (2005). In Colebrook and Sicilia (2007b) the location of undesirable facilities on multicriteria networks is looked into by using convex combinations of two objective functions. Some complexity analysis for the cent-dian location problem has been developped by Colebrook and Sicilia (2007a). Most approaches to the (in general NP-hard) multi-facilty case are treated as discrete location problems. Only recently Kalcsics et al. (2015) started looking into polynomial cases of multi-facility multicriteria location problems on networks.

Bibliography

Altay, N. and Green, W. G. (2006) "OR/MS research in disaster operations management". *European Journal of Operational Research*, vol. 175(1): 475–493.

Arriola, R., Ortega, F. A. and Pozo, M. A. (2009) "Aplicación de los problemas pickup and delivery en la gestión de emergencias de un sistema de ferrocarril". In *Actas del II MORE (II Jornadas sobre Modelos de Optimización Aplicados a la Planificación Robusta y a la Gestión de los Servicios de Transporte en Caso de Emergencia)*, vol. 1, pp. 55–64. Universidad de Sevilla.

Bunte, S. and Kliewer, N. (2009) "An overview on vehicle scheduling models". *Public Transport*, vol. 1: 299–317.

Carrizosa, E., Conde, E., Fernández, F. and Puerto, J. (1993) "Efficiency in euclidean constrained location problems." *Operations Research Letters*, vol. 14: 291–295.

Ceder, A. and Wilson, N. H. M. (1986) "Bus network design". *Transportation Research Part B: Methodological*, vol. 20(4): 331–344.

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

Chakroborty, P., Deb, K. and Sharma, R. K. (2001) "Optimal fleet size distribution and scheduling of transit systems using genetic algorithms". *Transportation Planning and Technology*, vol. 24(3): 209–226.

Colebrook, M. and Sicilia, J. (2007a) "A polynomial algorithm for the multicriteria cent-dian location problem". *European Journal of Operational Research*, vol. 179(3): 1008–1024.

Colebrook, M. and Sicilia, J. (2007b) "Undesirable facility location problems on multicriteria networks". *Computers & Operations Research*, vol. 34(5): 1491–1514.

Cominetti, R. and Correa, J. (1999) "The common-line problem in congested transit networks". *Transportation Science*, vol. 2001: 35–3.

Cominetti, R. and Correa, J. (2001) "Common-lines and passenger assignment in congested transit networks". *Transportation Science*, vol. 35(3): 250–267.

Daskin, M. (1995) *Network and Discrete Location: Models, Algorithms, and Applications.* Wiley, New York.

Daskin, M. S. (2013) *Network and discrete location: models, algorithms, and applications, 2nd edition.* Wiley, Hoboken.

Desaulniers, G. and Hickman, M. D. (2007) "Chapter 2 public transit". In C. Barnhart and G. Laporte, eds., *Transportation*, vol. 14 of *Handbooks in Operations Research and Management Science*, pp. 69–127. Elsevier.

Drezner, Z., ed. (1995) *Facility Location: A Survey of Applications and Methods.* Springer, New York.

Drezner, Z. and Hamacher, H., eds. (2002) *Facility Location: Applications and Theory.* Springer, New York.

Ehrgott, M. (2005) *Multicriteria Optimization.* Springer Berlin Heidelberg.

Ehrgott, M. and Gandibleux, X. (2000) "A survey and annotated bibliography of multiobjective combinatorial optimization". *OR Spectrum*, vol. 22: 425–460.

Fernández, E. and Puerto, J. (2003) "Multiobjective solution of the uncapacitated plant location problem". *European Journal of Operational Research*, vol. 145: 509–529.

Fosgerau, M. (2009) "The marginal social cost of headway for a scheduled service". *Transportation Research Part B: Methodological*, vol. 43(8–9): 813–820.

Gandibleux, X., Jaszkiewicz, A., Freville, A. and Slowinski, R. (2000) "Special issue in multiple objective metaheuristics". *Journal of Heuristics*, vol. 6: 291–431.

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

Grosfeld-Nir, A. and Bookbinder, J. (1995) "The planning of headways in urban public transit". *Annals of Operations Research*, vol. 60: 145–160.

Guihaire, V. and Hao, J.-K. (2008) "Transit network design and scheduling: A global review". *Transportation Research Part A: Policy and Practice*, vol. 42(10): 1251–1273.

Hakimi, S. (1964) "Optimum locations of switching centers and the absolute centers and medians of a graph". *Operations Research*, vol. 12(3): 450–459.

Hakimi, S. (1965) "Optimum distribution of switching centers in a communications network and some related graph theoretic problems". *Operations Research*, vol. 13(3): 462–475.

Hamacher, H. and Nickel, S. (1996) "Multicriteria planar location problems". *European Journal of Operational Research*, vol. 94(1): 66–86.

Hamacher, H. W., Labbé, M. and Nickel, S. (1999) "Multicriteria network location problems with sum objectives". *Networks*, vol. 33(2): 79–92.

Hamacher, H. W., Labbé, M., Nickel, S. and Skriver, A. J. (2002) "Multicriteria semi-obnoxious network location problems (msnlp) with sum and center objectives". *Annals of Operations Research*, vol. 110: 33–53.

Hamdouch, Y., Ho, H., Sumalee, A. and Wang, G. (2011) "Schedule-based transit assignment model with vehicle capacity and seat availability". *Transportation Research Part B: Methodological*, vol. 45(10): 1805–1830.

Hamdouch, Y., Szeto, W. and Jiang, Y. (2014) "A new schedule-based transit assignment model with travel strategies and supply uncertainties". *Transportation Research Part B: Methodological*, vol. 67(0): 35–67.

Hansen, P. (1980) "Bicriterion path problems". In *Multiple criteria decision making theory and application*, vol. 177 of Lecture Notes in Econom. and Math. Systems, pp. 109–127.

Kalcsics, J., Nickel, S., Puerto, J. and Rodríguez-Chía, A. M. (2015) "Several 2-facility location problems on networks with equity objectives". *Networks*, vol. 65(1): 1–9.

Laporte, G., Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2009) "Locating a metro line in a historical city centre: application to sevilla". *Journal of the Operational Research Society*, vol. 60(5): 1462–1466.

Laporte, G., Nickel, S. and Saldanha da Gama, F., eds. (2015) *Location Science*. Springer International Publishing Switzerland.

Larson, R. and Odoni, A. (1981) *Urban operations research*. Prentice-Hall: New York.

Liebchen, C., Schachtebeck, M., Schöbel, A., Stiller, S. and Prigge, A. (2010) "Computing delay resistant railway timetables". *Computers & Operations Research*, vol. 37(5): 857–868.

Lozano, A. J., Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2008) "Assessing the robustness in topological configurations of rapid transit systems". In *Proceedings of the XI ISOLDE (International Symposium on Locational Decisions)*, vol. 1, pp. 91–91. University of California in Santa Barbara.

Matisziw, T. C. and Murray, A. T. (2009) "Modeling s-t path availability to support disaster vulnerability assessment of network infrastructure". *Computers & Operations Research*, vol. 36(1): 16–26.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2008) "Robustez ante incidencias en el diseño topológico de redes metropolitanas de transporte público". In *Actas del VIII CIT (Congreso de Ingeniería del Transporte)*, vol. 1, pp. 1–8. Demarcación del Colegio de Ingenieros de Caminos, Canales y Puertos.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2009) "Effective allocation of fleet frequencies by reducing intermediate stops and short turning in transit systems". *Lecture Notes in Computer Science*, vol. 5868(8): 293–309.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2013) "A geometric model for an effective rescheduling after reducing service in public transportation systems". *Computers & Operations Research*, vol. 40(3): 737–746.

Mirchandani, P. and Francis, R. (1990) *Discrete Location Theory*. John Wiley and Sons, New York.

Murray, A. T. (2003) "A coverage model for improving public transit system accessibility and expanding access". *Annals of Operations Research*, vol. 123: 143–156.

Nickel, S. and Puerto, J. (2005) *Location Theory: A Unified Approach*. Springer-Verlag, Heidelberg, Germany.

Nickel, S., Puerto, J. and Rodríguez-Chía, A. M. (2005) "MCDM location problems". In J. A. Figueira, S. Greco and M. Ehrogott, eds., *Multiple Criteria Decision Analysis: State of the Art Surveys*, vol. 78 of *International Series in Operations Research & Management Science*, pp. 761–787. Springer, New York.

Nickel, S., Puerto, J. and Rodriguez-Chia, A. (2015) "Location problems with multiple criteria". In G. Laporte, S. Nickel and F. Saldanha da Gama, eds., *Location Science*, chap. 9, pp. 205–247. Springer International Publishing Switzerland.

Nuzzolo, A., Crisalli, U. and Rosati, L. (2012) "A schedule-based assignment model with explicit capacity constraints for congested transit networks". *Transportation Research Part C: Emerging Technologies*, vol. 20(1): 16–33. Special issue on Optimization in Public Transport+ISTT2011 Special issue on Optimization in Public Transport+International Symposium on Transportation and Traffic Theory (ISTTT), Berkeley, California, July 18–20, 2011.

Pel, A. J., Bel, N. H. and Pieters, M. (2014) "Including passengers' response to crowding in the dutch national train passenger assignment model". *Transportation Research Part A: Policy and Practice*, vol. 66(0): 111–126.

Poon, M. H., Wong, S. C. and Tong, C. O. (2004) "A dynamic schedule-based model for congested transit networks". *Transportation Research Part B: Methodological*, vol. 38(4): 343–368.

Puerto, J. (1996) "Lecturas en teoría de localización". Tech. rep., Universidad de Sevilla. Secretariado de Publicaciones.

Puerto, J. and Fernández, F. (1999) "Multicriteria minisum facility location problem". *Journal of Multi-Criteria Decision Analysis*, vol. 8: 268–280.

Ross, G. and Soland, R. (1980) "A multicriteria approach to the location of public facilities". *European Journal of Operational Research*, vol. 4: 307–321.

Schmöcker, J.-D., Bell, M. G. and Kurauchi, F. (2008) "A quasi-dynamic capacity constrained frequency-based transit assignment model". *Transportation Research Part B: Methodological*, vol. 42(10): 925–945.

Schöbel, A. (2005) "Locating stops along bus or railway lines–a bicriteria problem". *Annals of Operations Research*, vol. 136(1): 211–227.

Skriver, A. J., Andersen, K. A. and Holmberg, K. (2004) "Bicriteria network location (bnl) problems with criteria dependent lengths and minisum objectives." *European Journal of Operational Research*, vol. 156: 541–549.

Small, K. A. (1982) "The scheduling of consumer activities: Work trips". *The American Economic Review*, vol. 72(3): 467–479.

Spiess, H. and Florian, M. (1989) "Optimal strategies: A new assignment model for transit networks". *Transportation Research Part B: Methodological*, vol. 23(2): 83–102.

Törnquist, J. (2007) "Railway traffic disturbance management–an experimental analysis of disturbance complexity, management objectives and limitations in planning horizon". *Transportation Research Part A: Policy and Practice*, vol. 41(3): 249–266.

Ulungu, E. and Teghem, J. (1994) "Multi-objective combinatorial optimization problems: a survey". *Multi-Criteria Decision Analysis*, vol. 3: 83–104.

Van de Velde, D. (1999) "Organisational forms and entrepreneurship in public transport–part 1: Classifying organisational forms". *Transport Policy*, vol. 6: 147–157.

Vieira, G., Herrmann, J. and Lin, E. (2003) "Rescheduling manufacturing systems: A framework of startegies, policies, and methods". *Journal of Scheduling*, vol. 6: 39–62.

Warburton, A. (1983) "Quasiconcave vector maximization: connectedness of the sets of pareto-optimal and weak pareto-optimal alternatives." *Journal of Optimization Theory and Applications*, vol. 40: 537–557.

Wendell, R. and Hurter, A. (1973) "Location theory, dominance and convexity". *Operations Research*, vol. 21: 314–320.

Wirasinghe, S. (2003) *Advanced Modeling for Transit Operations and Service Planning*, chap. Initial planning for urban transit systems., pp. 1–29. Elsevier Science.

Wren, A. and Rousseau, J. (1995) *Computer-Aided Transit Scheduling*, chap. Bus driver scheduling–an overview, pp. 173–187. .

Yu, B., Yang, Z., Cheng, C. and Liu, C. (2005) "Optimizing bus transit network with parallel ant colony algorithm". In *Proceedings of the Eastern Asia Society for Transportation Studies*, vol. 5, pp. 374–389.

Zhang, F. and Melachrinoudis, E. (2001) "The maximin-maxisum network location problem". *Computational Optimization and Applications*, vol. 19(2): 209–234.

Chapter 1

Locating optimal timetables and vehicle schedules in a transit line.

ABSTRACT

This chapter deals with the Transit Network Timetabling and Scheduling Problem (TNTSP) in a public transit line. The TNTSP aims at determining optimal timetables for each line in a transit network by establishing departure and arrival times of each vehicle at each station. We assume that customers know departure times of line runs offered by the system. However, each user, traveling after or before than their desired travel time, will give rise to an inconvenience cost, or a penalty cost if that user cannot be served according to the scheduled timetable. The provided formulation allocates each user to the best possible timetable considering capacity constraints. The problem is formulated using a p-median based approach and solved using a clustering technique. Computational results that show useful applications of this methodology are also included.

Keywords: Timetabling; vehicle scheduling; schedule delay; location-allocation.

1.1 Introduction

The Transit Network Timetabling and Scheduling Problem (TNTSP) aims at determining optimal timetables for each line in a transit network by establishing departure and arrival times of each vehicle at each station. The TNTSP is based on the following general input: An infrastructure of a transport system described by a node set (network stations) and an edge set (tracks between adjacent stations), a trip demand matrix between pairs of nodes of the infrastructure, a set of transit lines with associated frequencies which have already been determined in order to satisfy such trip demand and, finally, a vehicle fleet with specific characteristics. The objective of the TNTSP consists of finding arrival and departure times of each vehicle at each station such that the demand satisfaction, required fleet size and vehicle capacities can be optimized/bounded.

The TNTSP integrates two stages of the global transit planning process, that is usually divided in a sequence of five steps (Ceder and Wilson, 1986): line planning, frequencies setting, timetabling, vehicle scheduling and crew rostering. Even separately, solving each of these problems implies a challenge in terms of computational complexity (Magnanti and Wong, 1984; Quak, 2003); however, a considerable amount of work points out the integration of several of these planning stages (Guihaire and Hao, 2008), in order to achieve interaction and feedback in the process as well as better quality results.

The TNTSP might be solved under different time contexts: for the strategic context (e.g. promoting the extension of the network infrastructure by means of including new line segments between stations), in the tactical planning (where regular vehicle scheduling for a given frequency -daily or weekly- is determined) and in real-time scenarios (where dispatchers must manage the traffic by making optimal decision about which vehicles to stop and where, according to data about vehicle positions that are continuously subject to an updating process).

Timetabling is the process of implementing the service frequency on each fixed route, providing arrival/departure times at each station. Timetables can be periodic (e.g. Liebchen et al., 2010) if they are repeated in time intervals. Although periodicity makes timetables easy-to-remember, non-periodic timetables (e.g. de Palma and Lindsey, 2001) can be implemented to more adequately fit within the current time-dependent demand pattern. Two different types of infrastructures can be analyzed to implement timetables: a single corridor (see, e.g., Brannlund et al., 1998, Caprara et al., 2002, Zhou and Zhong, 2007) or an entire network including transfers (see, e.g., Caprara et al., 2006). In both scenarios, the main objectives that are usually taken into consideration are addressed at maximizing the transfer synchronization, in order to minimize waiting times at transfers (e.g. Guo and Wilson, 2011) and minimizing the *schedule delay* (Small, 1982). The concept of schedule delay arises with the fact that arriving early is likely to involve some wasted time while for most users, arriving late has more severe repercussions. In this way, timetabling can be seen as a p-median problem (Hakimi, 1964) where the objective is to minimize the time/distance between passenger desired departure times and actual ones. Narrow headways lead users to arrive randomly to stations giving rise to a waiting cost for the user. On the other hand, when headways are wider, customers tend to strictly follow timetables, arriving only few minutes before the departure time. This last situation does not provoke a waiting cost for the user but an inconvenience cost to fit desired travel time to actual timetables (Grosfeld-Nir and Bookbinder, 1995; Fosgerau, 2009).

The vehicle scheduling problem consists of allocating a set of vehicles to a set of timetables, taking into consideration some practical requirements like depot/s location/s, vehicle features (speed, size, maintenance costs, fuel costs) and other extensions. An optimal schedule minimizes the fleet size as well as operational costs. Vehicle scheduling problems have received considerable attention in the literature considering a single corridor (see, e.g., Higgins et al., 1996, Oliveira and Smith, 2000, Zhou and Zhong, 2005) and an entire network (Cai and Goh, 1994, Chew et al., 2001). A recent overview in vehicle scheduling problem has been provided by Bunte and Kliewer (2009).

Dealing with the scheduling problem separately from the timetabling problem implies that fleet size requirements cannot be bounded, only minimized. Moreover, if timetables are too plentiful for a small fleet size, a feasible vehicle schedule cannot even be guaranteed. In this sense, little work has been developed searching for the integration of timetabling and vehicle scheduling (Guihaire and Hao, 2010). One of the first approaches that integrates both timetabling and vehicle scheduling problem is the one provided by Chakroborty et al. (2001), where periodic timetables are determined in order to minimize waiting times at transfer stations as well as the required fleet size. Castelli et al. (2004) deal with non periodic timetables assuming that routes, means of transport and quality of services are fixed in advance. The operator's main objective is to minimize their costs, while serving, at the same time, as many customers as possible. The solution procedure schedules a single line at a time, possibly re-optimizing or correcting the previous decisions at each step. Chang and Chung (2005) consider a single, one-way track train timetabling problem for a rapid transit system. Liu and Shen (2007) integrate timetabling and vehicle scheduling by using a bilevel formulation, where the upper level is referred to the operators' objectives while the lower level reflects the user interests. Guihaire and Hao (2010) develop an integrated approach without considering periodic timetables but they include evenness of the line headways as one of the optimization criterion. Finally, Cadarso and Marín (2012) integrate railway timetabling and scheduling updating the frequencies known from the railway line planning problem. Frequencies for some arcs are maintained in a determined frequency window as well as headways that are maintained for every train line in the network.

This chapter deals with a customer-oriented timetabling-scheduling model applied to the setting of a public transit corridor with two lines (one per each direction). A pre-set number of line runs (vehicle expeditions along a line) will be located in each line along the time horizon, under constraints determined by the fleet size and the maximum number of allowed line runs. Additionally, we will assume non-periodic timetables that will be known by users in advance. Therefore, timetables will be determined in order to optimally serve the existing demand by considering that those individuals, traveling later than or ahead of their desired travel time, will suffer a user's inconvenience cost. Customers are supposed to suffer an inconvenience cost, if their preferred pickup/delivery times vary from the actual ones, and a penalty cost, if the requests are not served in a time window. In this sense, the user behavior, explained by means of inconvenience costs, is motivated from the contexts of the vehicle routing problem with time windows (Cordeau et al., 2007b) and transportation on demand (Cordeau et al., 2007a), where disaggregated demand and the human factor acquire great significance. To the best of our knowledge, inconvenience costs have not been considered in the TNTSP literature. Additionally, since we are dealing with a public transportation system where users choose freely in which vehicle they want to board, the situation where users are freely allocated to vehicles, ignoring capacity levels, will be studied.

The remainder of this chapter is structured as follows. Section 1.2 is devoted to describing the context where the model is formulated. In Section 1.3 a variation of the p-median problem is proposed to optimally locate a number of line runs ensuring vehicle schedules. In order to decrease the size of the problem, a clustering algorithm is described in Section 1.4. Computational experiments are provided in Section 1.5 in order to show the usefulness and applicability of this methodology. The extension of this methodology to transportation networks composed of several transit lines is described in Section 1.6. Finally, conclusions will be described in Section 1.7.

1.2 Problem description

1.2.1 Infrastructure

Let L be a transit corridor consisting of a node set S (stations) and an edge set E (tracks). Let $l \in \mathcal{L}$ be the set of feasible lines in L. For the setting considered in this chapter, $\mathcal{L}$ only will contain two lines; that is, $\mathcal{L} = \{1, 2\}$, where $l = 1$ is a directed transit line running along L and $l = 2$ is also running along L but in the opposite direction. We denote by $\langle s, l\rangle$ the station in position $s \in \{1, ..., |S_l|\}$ belonging to the set of stations $S_l \subseteq S$ of a given line $l \in \mathcal{L}$. Additionally, let E_l be the subset of E that contains all edges used by line l. Each identical vehicle will operate along L during a time horizon that will be discretized into a set of time slots $t \in T = \{1, ..., |T|\}$. Each vehicle performs a number of line runs or expeditions along a line. Line runs have to be located in time for each line. The total number of line runs to locate in line l (ρ_l), the vehicle capacity (Q) and the fleet size κ of each line will be assumed to be input data of the problem.

1.2.2 Demand

Let I be the set of transportation requests formulated by customers of transit corridor L. Each request $i \in I$ involves the following information:

1. A pair of origin and destination stations, denoted by $\langle s_i, l_i\rangle$ and $\langle s'_i, l_i\rangle$, respectively. Such stations must be associated to a line l_i with edges $e \in E_{l_i}$ that will be used as a path to satisfy request i. With this, it can be defined a parameter m_{ie} equal to one if edge $e \in E_{l_i}$ is used when request $i \in I$ is served or equal to zero otherwise.
2. A preferred departure time t_i to locate a line run in station $\langle 1, l_i\rangle$. Furthermore, $t_i \in [t_i^-, t_i^+]$, which denotes the earliest and the latest times that are admissible for serving request i.
3. A penalization cost c_i that will be paid if the corresponding request is not served.
4. Trips along the transit line are direct and travel times are known in advance, so the users' inconvenience with respect to the arrival time can be obviated. Denoting by $T_i = \{t_i^-, t_i^- + 1, ..., t_i^+\}$ the set of feasible time slots where a line run can be located in order to serve request i, parameter φ_{it} will compute the relative cost of allocating request i to the line run which departs from station $\langle 1, l_i\rangle$ at time $t \in T_i$. The total inconvenience cost of a transportation request is defined as $c_i\varphi_{it}$ if i is allocated to a line run in t, or c_i if i is not allocated to any line run.
5. Following previous definitions, I_{lt} denotes the subset of requests that can be served locating a line run for line l in the time slot t, that is $I_{lt} = \{i \in I : t \in T_i \wedge l_i = l\}$.

1.2.3 Timetables and vehicle schedules

The concept of timetable must be formalized as follows. Given the set of line runs $r \in R_l$ defined in line l, with $|R_l| = \rho_l$, a timetable Θ along partition T is defined as the set of arrival/departure times at each station for each line run: $\Theta = \{(\theta^+_{\langle s,l\rangle r}, \theta^-_{\langle s,l\rangle r}), l \in \mathcal{L}, \langle s,l\rangle \in S_l, r \in R_l\}$.

Potentially, all timetables can be generated over the three defined sets. However, the number of feasible timetables can be highly reduced by means of the following result:

Property 1.1- Assuming that:

1. stopping time at each station is known/prefixed for all line runs:
   $\theta^-_{\langle s,l\rangle r} - \theta^+_{\langle s,l\rangle r} = \lambda_{\langle s,l\rangle}, \quad l \in \mathcal{L}, \langle s,l\rangle \in S_l, r \in R_l$
2. travel times between consecutive stations is known/prefixed for all line runs:
   $\theta^+_{\langle s+1,l\rangle r} - \theta^-_{\langle s,l\rangle r} = \mu_{\langle s,l\rangle}, \quad l \in \mathcal{L}, \langle s,l\rangle \in S_l : s < |S_l|, r \in R_l$

the following properties can be stated:

- a fixed travel time τ_l can be assumed in order to complete a line run:
  $\theta^+_{\langle |S_l|,l\rangle r} - \theta^-_{\langle 1,l\rangle r} = \tau_l, \quad l \in \mathcal{L}, r \in R_l$
- timetables can be redefined as follows:
  $\Theta \equiv x = \{x_{lt}, l \in \mathcal{L}, t \in T\}$
  where $x_{lt} \in \{0,1\}$ is equal to 1 if and only if a line run departs from the first station of line l at time t.

Note that the number of vehicles required to perform the timetable cannot be greater than the fleet size, even when a vehicle can perform several line runs in the time period under consideration. For this reason, we must characterize when a timetable can be performed by the given fleet size.

Definition 1.1- A timetable x is a feasible κ-vehicle schedule if the number of vehicles required to perform x is less or equal than the fleet size κ of each line, thus

$$0 \leq \sum_{t'=1}^{t} x_{1t'} - \sum_{t'=1}^{t-\tau_2} x_{2t'} \leq \kappa \qquad t \in T \tag{1.1}$$

$$0 \leq \sum_{t'=1}^{t} x_{2t'} - \sum_{t'=1}^{t-\tau_1} x_{1t'} \leq \kappa \qquad t \in T. \tag{1.2}$$

This definition ensures that between time slots $t = 1$ and $t = \tau_l$ no more than κ vehicles leave the depot of any line. Note that when $t < \tau_2$, the second sum in (1.1) does not count any term. After $t = \tau_l$, vehicles that arrive to the end of one line can be used by the other, but at any moment the difference between those vehicles which have left a depot and those which have arrived to such a depot cannot be negative nor greater than κ. This will lead us to a characterization between an optimal timetable and an optimal vehicle schedule.

Property 1.2- Assuming that before starting a line run, vehicles are always empty, they can remain stopped as long as necessary without any additional cost. Therefore, an optimal timetable x satisfying (1.1) and (1.2) is also an optimal vehicle schedule.

1.3 Formulation

1.3.1 The ρ-median problem

Two sets of binary variables are considered in the formulation:

$x_{lt} \in \{0,1\}$ equal to 1 when a vehicle starts a line run in line l at time t

$y_{it} \in \{0,1\}$ equal to 1 when request i is allocated to a vehicle which starts a line run at time t.

The mathematical model that describes our problem is defined as follows:

$$z = \min \sum_{i \in I} [\sum_{t \in T_i} c_i \varphi_{it} y_{it} + \sum_{t \in T_i} c_i (1 - y_{it})] \tag{1.3a}$$

$$\text{s.t.:} \quad \sum_{t \in T} x_{lt} \leq \rho_l \qquad l \in \mathcal{L} \tag{1.3b}$$

$$\sum_{t \in T_i} y_{it} \leq 1 \qquad i \in I \tag{1.3c}$$

$$y_{it} - x_{l_i t} \leq 0 \qquad i \in I, t \in T_i \tag{1.3d}$$

$$0 \leq \sum_{t'=1}^{t} x_{1t'} - \sum_{t'=1}^{t-\tau_2} x_{2t'} \leq \kappa \qquad t \in T \tag{1.3e}$$

$$0 \leq \sum_{t'=1}^{t} x_{2t'} - \sum_{t'=1}^{t-\tau_1} x_{1t'} \leq \kappa \qquad t \in T \tag{1.3f}$$

$$\sum_{i \in I_{lt}} y_{it} m_{ie} \leq Q x_{lt} \qquad l \in \mathcal{L}, e \in E_l, t \in T \tag{1.3g}$$

$$x_{lt} \in \{0,1\} \qquad l \in \mathcal{L}, t \in T \tag{1.3h}$$

$$y_{it} \in \{0,1\} \qquad i \in I, t \in T_i. \tag{1.3i}$$

Objective function (1.3a) minimizes the total users' inconvenience. For the sake of understandability, we present (1.3a) as defined in Section 1.2.2. Constraint (1.3b) ensures that no more than ρ_l line runs are located in line l. Constraint (1.3c) guarantees that request i is not allocated to more than one line run, avoiding negative terms in the second part of the objective function. Constraint (1.3d) ensures that requests can only be allocated to time slots where a line run has been located. Constraints (1.3e) and (1.3f) forces that no more than κ vehicles are used on each line. Constraint (1.3g) guarantees that no more than Q requests use edge e of line l in a line run located at time t.

The provided formulation exhibits significant similarities with the classic p-median problem (Hakimi (1964, 1965)), where the total sum of weighted distances between a given set of customers, and a set of locations for potential facilities is minimized. In our formulation, facilities are line-runs, customers are travelers and distances are measured by scheduled delays (we recall Section 1; Small, 1982). In our model, a request i can be refused in exchange for the penalty cost c_i, and deviations from desired travel times are penalized by using the discrete parameter φ_{it}. Additionally, constraints for controlling schedules and capacities have been stated.

1.3.2 Public context for the ρ-median problem

Since the access to vehicles is performed inside stations, all the infomation regarding timetables is availlable for passengers so they are free to choose between those line runs that better fit into their own interests. This idea contrasts with other transport systems where the operator is the unique decision maker who can choose which request should be served, when and using which vehicle. In order to emphasize the importance of the passenger point-of-view in the modelization, additional constraints can be added. First, we impose that a passenger must be allocated to a line run if there is at least one line run located in T_i:

$$\sum_{t \in T_i} x_{l_i t} \leq \rho_{l_i} \sum_{t \in T_i} y_{it} \qquad i \in I. \tag{1.3j}$$

Additionally, constraint (1.3k) ensures that each request is allocated to a line run if and only if such line run is the one with lowest inconvenience cost in T_i:

$$\sum_{\substack{t' \in T_i \\ \varphi_{it'} > \varphi_{it}}} y_{it'} + x_{l_i t} \leq 1 \qquad i \in I, t \in T_i. \tag{1.3k}$$

Constraint (1.3k) is adapted from Wagner and Falkson (1975) and belongs to the so called closest assignment constraints. According to Espejo et al. (2012), this kind of constraints can be modelled in many different ways, giving rise to better/worse linear programming relaxations. As shown in that paper, the constraint provided by Wagner and Falkson (1975) can be strengthened by using the fixed numbers of facilities (line runs) to locate. Even with such improvements, we must note that the inclusion of this constraint together with (1.3j) considerably increases the computational complexity of the proposed model. In order to provide good solutions within a reasonable time of computation, a request clustering algorithm is introduced in the next section in order to reduce the number of variables and constraints of the model.

1.4 Clustering algorithm

The number of requests $|I|$ can be highly reduced considering subsets of requests $\bar{I}_{lt} \subseteq I$ with a common preferred departure time for a line run, thus $\bar{I}_{lt} = \{i \in I : l_i = l \wedge t_i = t\})$. We define a new set of clustered requests $j \in J$ in such a way that each subset $\bar{I}_{lt}$ is identified with a transportation request j weighted by a factor q_j. In this way, we can solve the problem by using the set $j \in J$ instead of $i \in I$. Note that clustered requests can be built if $|I| > |T|$ or if $\exists\, l \in \mathcal{L}, t \in T$ such that $|\bar{I}_{lt}| > 1$. In order to define the clustering algorithm, we require the following parameters:

- $\bar{t}_j$: preferred departure time for clustered request j in order to locate a line run in station $\langle 1, l_j \rangle$
- $\bar{c}_j$: penalization cost that will be paid if the corresponding clustered request is not served
- $\bar{m}_{je}$: number of requests in I that are grouped in j and use edge e (we denote $m_i = [m_{i1}, m_{i2}, ..., m_{i|E_{l_i}|}]$ and $\bar{m}_j = [\bar{m}_{j1}, \bar{m}_{j2}, ..., \bar{m}_{j|E_{l_j}|}]$)
- q^{max}: maximum weight for a clustered request, that is $q_j \leq \min\{q^{max}, Q\}$.

Algorithm 1 describes how these parameters are constructed.

Algorithm 1: Clustering requests

input : Requests $i \in I$, clusters $J := \{\,\}$, parameters c_i, m_{ie} and maximum weight q^{max} for a clustered request

output: Clustered requests $j \in J$, and parameters $\bar{t}_j, \bar{c}_j, \bar{m}_j$

for *each request* $i \in I$ **do**
    if $\nexists\, j \in J$ *clustered request such that* $\bar{t}_j = t_i$ *and* $q_j < q^{max}$ **then**
        Create the clustered request $j = |J| + 1$: $[J, q_j, \bar{c}_j, \bar{m}_j] := [J \cup \{j\}, 1, c_i, m_i]$;
        else
            Update $[q_j, \bar{c}_j, \bar{m}_j] := [q_j + 1, \bar{c}_j + c_i, \bar{m}_j + m_i]$

For each request, we have to check if there exists a clustered request such that $\bar{t}_j = t_i$ and $q_j \leq q^{max}$. Since we can obviate searching along requests $j \in J$ such that $q_j = q^{max}$, in the worst case we check as many requests as time slots in T. Thus, the complexity of Algorithm 1 is $O(|I||T|)$.

Next, in order to solve the ρ-median problem with clustered requests (c-ρ-median problem), we apply the following procedure:

Procedure 1: (c-ρ-median problem)

1. Solve (1.3a)–(1.3k) considering the set of clustered requests J instead of I. Let x' be the timetable solution returned.
2. Solve (1.3a)–(1.3k) considering the set of requests I and fixing x variables as $x := x'$ (thus, only variables y and z have to be found). Let z^{UB} be the best (smallest) upper bound

returned and let y' be the request assignment. The solution given by the c-ρ-median problem is $\{z = z^{UB}, x = x', y = y'\}$.

3. Let z^{LB} be the best lower bound returned by the ρ-median problem. The gap induced by the c-ρ-median problem can be obtained by means of

$$c\text{-}gap = \frac{100(z^{UB} - z^{LB})}{z^{LB}}.$$

Note that without considering capacity constraints, optimal solutions for the c-ρ-median and ρ-median problems are equivalent. On the other hand, under capacity constraints clusters cannot be split and only complete clusters can be allocated/rejected. Consequently, requests within a cluster cannot be allocated to different line-runs with equal inconvenience costs. Therefore, the solution provided by the c-ρ-median problem is an upper bound of the ρ-median problem.

1.5 Computational experience

In order to show the applicability of the previous model and algorithm, a scenario composed of one transit corridor with two lines (each one running on a different direction) along eight stations has been considered. A random instance of $|I| = 1000$ requests has been generated in the time interval with desired arrival times following a uniform probability distribution. All requests have been assumed to have equal penalization costs c_i and inconvenience costs (φ_{it}) defined in $T_i = \{\max\{0, t_i - 4\}, ..., \max\{0, t_i - 1\}, t_i, \min\{|T|, t_i + 1\}, ..., \min\{|T|, t_i + 4\}\}$ that is, $|T_i| \simeq 9$ options for each $i \in I$. The discretized inconvenience function (see Figure 1.1) has been taken as follows:

$$\varphi_{it} = \min\left\{1, \left(\frac{\max\{0, t_i - t\}}{\max\{1, t_i - t_i^-\}}\right)^2 + \left(\frac{\max\{0, t - t_i\}}{\max\{1, t_i^+ - t_i\}}\right)^2\right\}.$$

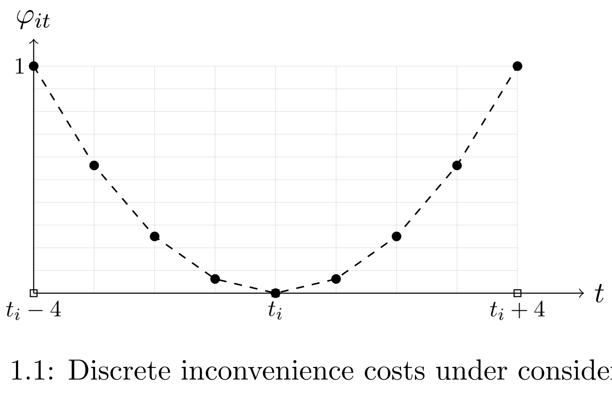


Figure 1.1: Discrete inconvenience costs under consideration.

The time horizon has been split into 60 time slots ($|T| = 60$) of standardized length $\|t\|$ (for instance, $\|t\| = 1$ and $|T| = 60$ implies a time horizon of 1 hour whereas $\|t\| = 2$ and $|T| = 60$ implies a time horizon of 2 hours). Distances (in time) between stations are considered equal to $4.5\|t\|$ and a time for boarding and alighting equal to $0.5\|t\|$ will be required. Fleet sizes (κ) will be assumed equal to 2, 3 and 4. Line runs will vary in ρ=1, 2, 3, 4, 5, 6, 8, 10, 12, that would provide in a time interval of $60\|t\|$ an approximate frequency of 60, 30, 20, 15, 12, 10, 7.5 and 5 time (measured in time units of length $\|t\|$).

The computational experience also includes different scenarios depending on the capacity requirements established. Scenario $S1$ assumes the version of the problem without capacity constraints. In scenarios $S2_{(Q)}, Q \in \{40, 45\}$ constraint (1.3g) is activated for establishing a version of selective capacity in the problem. In scenarios $S3_{(Q)}, Q \in \{40, 45\}$ constraints (1.3j) and (1.3k) are activated for establishing the optimal timetable choices for single requests. Finally, scenarios $S3_{(Q)}^{(q^{max})}, Q \in \{40, 45\}, q^{max} \in \{3, 5\}$ show that constraints (1.3j) and (1.3k) are activated and requests have been clustered considering a maximum cluster size equal to q^{max}.

All instances have been solved using ILOG CPLEX 12.2 on a personal computer with an Intel(R) Core(TM)i7 CPU 3.4 GHz processor and 16 GB RAM. Default solver values were used for all parameters.

		$S1$		$S2_{(45)}$		$S3_{(45)}$		$S2_{(40)}$		$S3_{(40)}$	
κ	ρ	θ_1	θ_2	θ_1	θ_2	θ_1	θ_2	θ_1	θ_2	θ_1	θ_2
2	1	18.27	29.2	16.26	22.4	9.97	15.4	15.7	21.8	6.6	13.1
2	2	35.97	55.5	31.72	42.6	25.87	35.3	30.39	40.8	23.26	31.1
2	3	49.13	73.2	44.71	58.2	34.35	49.1	42.81	55.4	28.03	40.9
2	4	49.13	73.2	46.01	59.2	36.55	51	45.02	56.4	28.03	40.9
2	5	49.13	73.2	46.01	59.2	36.55	51	45.02	56.4	28.03	40.9
2	6	49.13	73.2	46.01	59.2	36.55	51	45.02	56.4	28.03	40.9
2	8	49.13	73.2	46.01	59.2	36.55	51	45.02	56.4	28.03	40.9
2	10	49.13	73.2	46.01	59.2	36.55	51	45.02	56.4	28.03	40.9
2	12	49.13	73.2	46.01	59.2	36.55	51	45.02	57.1	28.03	40.9
3	1	18.27	29.2	16.26	22.4	9.97	15.4	15.7	21.8	6.6	13.1
3	2	35.97	55.5	31.72	42.6	25.87	35.3	30.39	40.8	23.26	31.1
3	3	52.4	81	46.89	61.4	37.84	50.2	44.84	58	35.19	44.3
3	4	66.38	91.3	59.73	76.4	45.57	61	57.11	72.7	39.92	54.2
3	5	66.38	91.3	61.29	78.9	48.83	62.7	59.55	74.9	42.57	52.8
3	6	66.38	91.3	61.29	78.9	48.34	59.5	59.55	74.9	43.72	52.5
3	8	66.38	91.3	61.29	78.9	48.76	65.6	59.55	74.9	42.91	52.1
3	10	66.38	91.3	61.29	78.9	48.6	63.6	59.55	74.9	43.72	52.5
3	12	66.38	91.3	61.29	78.9	48.6	63.6	59.55	74.9	43.33	52.1
4	1	18.27	29.2	16.26	22.4	9.97	15.4	15.7	21.8	6.6	13.1
4	2	35.97	55.5	31.72	42.6	25.87	35.3	30.39	40.8	23.26	31.1
4	3	52.4	81	46.89	61.4	37.84	50.2	44.84	58	35.19	44.3
4	4	68.09	97.5	61.08	77.2	47.02	59.1	58.37	73	41.11	54
4	5	79.59	100	73.7	89.7	56.08	70.8	70.77	85.2	48.61	61.5
4	6	79.94	100	75.33	92.1	63.15	81.7	73.16	88.8	52.23	60.1
4	8	79.94	100	75.33	92.1	60.33	75.5	73.16	88.8	53.77	69.8
4	10	79.94	100	75.33	92.1	60.34	74.8	73.16	88.8	53.04	60.9
4	12	79.94	100	75.33	92.1	61.48	80.5	73.16	88.8	53.88	63.7

Table 1.1: Demand coverage for the different scenarios.

Table 1.1 shows the comparative objective values obtained for different scenarios, varying the fleet size, line runs and vehicle capacities. For the sake of favoring the comparison of results arising from heterogeneous contexts, the initial objective value z has been normalized and complemented by means of the function $\theta_1 \equiv 100(1 - z/|I|)$. Moreover, the percentage of covered requests with any level of satisfaction has been also included by means of the function $\theta_2 \equiv 100(1 - (\sum_i \sum_{|T_i|}) y_{it}/|I|)$. Logically, the inconvenience perception determines the level of satisfaction attained, holding in all scenarios $\theta_2 \geq \theta_1$. Additionally, the coverage level increases as long as Q, ρ or κ increase. However, θ_1 and θ_2

cannot improve at a certain level without including more vehicles. Irregularities in objective values for scenarios $S3_{(45)}$ and $S3_{(40)}$ at $\kappa = 4$, $\rho = 6, 8, 10, 12$ are due to the fact that optimality is not reached before the time limit (3600 seconds) under consideration.

Tables 1.2 and 1.3 show running times in seconds (*sec*) and gaps (*gap*, *c-gap*) obtained for the different scenarios. We denote as *gap*, the relative gap computed with the best lower bound obtained by Cplex within a time limit of 3600 seconds and the best upper bound obtained by Cplex within a time limit of 3600 seconds or 1200 seconds (indicated in the table). Additionally, *c-gap* denotes the gap induced by the clustering algorithm as described in Section 1.4 taking as reference the best upper bound provided by Cplex for $S3_{(Q)}$ within a time limit of 3600s.

		time limit=3600s				time limit=1200s					
		$S2_{(45)}$		$S3_{(45)}$		$S3_{(45)}$		$S3^{(3)}_{(45)}$		$S3^{(5)}_{(45)}$	
κ	ρ	*sec*	*gap*	*sec*	*gap*	*sec*	*gap*	*sec*	*c-gap*	*sec*	*c-gap*
2	1	4	0	0	0	0	0	0	0	0	0
2	2	2.3	0	60.6	0	59.8	0	9.1	0	18.4	0
2	3	2	0	301.1	0	300.9	0	36.9	0	31.3	0
2	4	2.5	0	1946.6	0	1200	0	28	0	14.8	0
2	5	2.5	0	308.6	0	308	0	79.2	0	46	0
2	6	2.5	0	1329.3	0	1200	0	36.9	0	44.8	0
2	8	2.5	0	1088.9	0	1090	0	75.5	0	61.6	0
2	10	2.6	0	2656.8	0	1200	0.01	30.6	0	29.4	0
2	12	2.5	0	1307.2	0	1200	0	30.7	0	29.2	0
3	1	4	0	0	0	0	0	0	0	0	0
3	2	2	0	68.6	0	68.7	0	10.7	0	24.5	0
3	3	2.4	0	2141.1	0	1200	0	141.1	0	23.6	0
3	4	2.8	0	3600	1.73	1200	1.99	412.3	1.99	223.8	1.99
3	5	2.9	0	3600	1.52	1200	1.88	328.7	1.77	109.5	1.77
3	6	2.7	0	3600	1.92	1200	2.92	166.3	2.23	118.6	2.23
3	8	3.1	0	3600	1.42	1200	1.7	369.6	1.62	244	1.62
3	10	3	0	3600	1.05	1200	1.2	277.6	1.09	120.1	1.09
3	12	2.8	0	3600	1.88	1200	2.31	327.2	2.22	295.9	2.22
4	1	3.8	0	0	0	0	0	0	0	0	0
4	2	2	0	380.7	0	381	0	11.5	0	5	0
4	3	2.4	0	2527.5	0	1200	0.06	184.4	0	60.7	0
4	4	3.9	0	3600	0.63	1200	0.87	441.8	0.65	464.8	0.65
4	5	3.3	0	3600	2.72	1200	4.34	1200	4.08	1200	3.81
4	6	3.7	0	3600	2.61	1200	5.14	1200	3.9	1200	4.24
4	8	3.6	0	3600	3.15	1200	5.98	1200	3.89	1200	4.3
4	10	3.9	0	3600	3.2	1200	5.13	1200	3.9	427	3.98
4	12	3.8	0	3600	2.84	1200	5.56	1200	3.99	242.8	3.68

Table 1.2: Running times and gaps obtained for the different scenarios and $Q = 45$.

Note first that we do not report running times or gaps for $S1$ since all instances were solved to optimality in less than 0.2 seconds. Even $S2_{(Q)}$ can be solved for each instance in a few seconds. We remark that theoretical scenarios $S1$ and $S2$ are interesting for establishing a comparative analysis on values reached in the objective function. On the other hand, scenario $S3$ is closer to the real operability although obtaining optimal solutions for that context requires a considerable computational effort. In order to assess the time consumed within the computation of optimal solutions for scenario $S3$, different scenarios for maximum running times of 3600 seconds and 1200 seconds have been analyzed. As mentioned in the introduction (third paragraph), the TNTSP might be called to solve decision problems in contexts characterized by strict limitations of time, going from those where speed might

		time limit=3600s				time limit=1200s					
		$S2_{(40)}$		$S3_{(40)}$		$S3_{(40)}$		$S3^{(3)}_{(40)}$		$S3^{(5)}_{(40)}$	
κ	ρ	*sec*	*gap*	*sec*	*gap*	*sec*	*gap*	*sec*	*c-gap*	*sec*	*c-gap*
2	1	3.3	0	0	0	0	0	0	0	0	0
2	2	2.3	0	62	0	61.9	0	14.7	0	5.6	0
2	3	2.5	0	206.4	0	206.4	0	34.8	0	20	0
2	4	3	0	1186.4	0	1190.6	0	85.5	0	41.3	0
2	5	2.9	0	561.1	0	560.5	0	119.5	0	31.8	0
2	6	2.7	0	190.6	0	190.6	0	71.8	0	37.9	0
2	8	2.9	0	846.3	0	846.8	0	102.4	0	16.7	0
2	10	3.7	0	248.4	0	247.9	0	91.8	0	39.3	0
2	12	3.6	0	2017.3	0	1200	0	92.2	0	39.5	0
3	1	3.1	0	0	0	0	0	0	0	0	0
3	2	2.3	0	61.3	0	61.7	0	15	0	5.9	0
3	3	2.7	0	967.8	0	967.9	0	72	0	30.8	0
3	4	3.6	0	3600	1.42	1200	1.72	443.9	1.57	91.2	1.57
3	5	5.4	0	3600	2.06	1200	3.3	316.5	2.33	118.9	2.33
3	6	4.5	0	3600	1.52	1200	2.1	210.8	1.8	213.5	1.8
3	8	4.9	0	3600	1.35	1200	1.86	1051.7	1.49	132.4	1.49
3	10	5.1	0	3600	1.56	1200	2.73	537.9	1.92	171.4	1.92
3	12	5	0	3600	1.43	1200	1.75	176.6	1.62	153.4	1.62
4	1	3.1	0	0	0	0	0	0	0	0	0
4	2	2.3	0	67.1	0	67	0	11	0	6	0
4	3	2.5	0	1239.4	0	1200	0	237.8	0	35.8	0
4	4	3	0	3581.9	0	1200	0.14	365.3	0	173.9	0
4	5	4.2	0	3600	3.1	1200	4.87	1200	4.55	198	4.55
4	6	5.7	0	3600	3.72	1200	5.8	1200	5.22	271.9	5.22
4	8	5.9	0	3600	3.5	1200	5.48	1200	5.11	659.1	5.4
4	10	6.5	0	3600	3.54	1200	5.53	1200	4.72	542.4	5.23
4	12	6.7	0	3600	3.42	1200	5.53	1200	4.72	1050.3	5.37

Table 1.3: Running times and gaps obtained for the different scenarios and $Q = 40$.

not be important to those where a fast solution might be required.

Under a time limit of 3600 seconds Cplex is able to solve to optimality all instances for $\kappa = 2$ leaving small gaps for $\kappa = \{3, 4\}$. Note that these gaps are due to the effect of capacities and closest assignment. Decreasing the time limit to 1200 seconds the gaps increase around a 2%. Under a time limit of 1200 seconds, the resolution of $S3^{(3)}_{(Q)}$ reaches optimality in all instances for $\kappa = 2$ increasing slightly the gaps of $S3_{(Q)}$ after 3600 seconds for $\kappa = \{3, 4\}$. However, $S3^{(3)}_{(Q)}$ outperforms $S3_{(Q)}$ under a time limit of 1200 seconds in terms of gap and running time. The same observation can be made for $S3^{(5)}_{(Q)}$ with lower running times.

In conclusion, Tables 1.2 and 1.3 prove the usefulness of introducing the clustering algorithm in scenario $S3$, because running times become considerably reduced in exchange for a small increase of the gap and, sometimes, even improves the actual ones. This methodology is of special interest for solving bigger size instances not considered in this chapter. In addition cluster sizes allow an adaptation of the procedure in accordance with the instance size.

Summarizing, Table 1.4 provides the results obtained for Scenario 3. In this table, we include averages and maximum values of computational times ($\overline{sec}$, sec^*), gaps ($\overline{gap}$, gap^*) and clustering gaps ($\overline{c\text{-}gap}$, $c\text{-}gap^*$) computed for different values of κ and all values of ρ.

κ		time limit=3600s $S3_{(45)}$	$S3_{(40)}$	time limit=1200s $S3_{(45)}$	$S3_{(40)}$		time limit=1200s $S3^{(3)}_{(45)}$	$S3^{(3)}_{(40)}$	$S3^{(5)}_{(45)}$	$S3^{(5)}_{(40)}$
2	$\overline{sec}$	999.9	590.94	728.74	500.52	$\overline{sec}$	36.32	68.08	30.61	25.79
	sec^*	2656.8	2017.3	1200	1200	sec^*	79.2	119.5	61.6	41.3
	$\overline{gap}$	0	0	0	0	$\overline{c\text{-}gap}$	0	0	0	0
	gap^*	0	0	0.01	0	$c\text{-}gap^*$	0	0	0	0
3	$\overline{sec}$	2645.52	2514.34	940.97	914.4	$\overline{sec}$	225.94	313.82	128.89	101.94
	sec^*	3600	3600	1200	1200	sec^*	412.3	1051.7	295.9	213.5
	$\overline{gap}$	1.06	1.04	1.33	1.5	$\overline{c\text{-}gap}$	1.21	1.19	1.21	1.19
	gap^*	1.92	2.06	2.92	3.3	$c\text{-}gap^*$	2.23	2.33	2.23	2.33
4	$\overline{sec}$	2723.13	2543.16	975.67	940.78	$\overline{sec}$	737.52	734.9	533.37	326.38
	sec^*	3600	3600	1200	1200	sec^*	1200	1200	1200	1050.3
	$\overline{gap}$	1.68	1.92	3.01	3.04	$\overline{c\text{-}gap}$	2.27	2.7	2.3	2.86
	gap^*	3.2	3.72	5.98	5.8	$c\text{-}gap^*$	4.08	5.22	4.3	5.4

Table 1.4: Summary of the results obtained for Scenario 3

1.6 Extension to transportation networks composed of several transit lines

Previous sections describe how to jointly plan timetables and vehicle schedules along a single transit line for potential customers traveling between origin and destinations of such a line. The extension of the scenario of a single transit line to the more general case where multiple lines are considered in the network can be performed including two new kinds of transportation requests coming from previous lines or going towards other lines.

- **Case 1:** Let i be a transportation request going from $\langle s_i, l_i\rangle$ towards $\langle s'_i, l_i\rangle$. If we assume that i requires to transfer to a second line at $\langle s'_i, l_i\rangle$, then t_i is the time slot that minimizes the waiting time at $\langle s'_i, l_i\rangle$. However, i cannot arrive to $\langle s'_i, l_i\rangle$ later than the departure time of the second vehicle. Thus, the inconvenience costs of this kind of request is maximum after t_i as depicted in Figure 1.2.

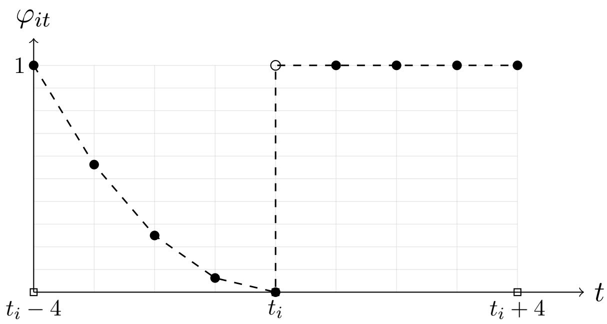


Figure 1.2: Inconvenience costs for a transportation request i, going from station $\langle s_i, l_i\rangle$ towards $\langle s'_i, l_i\rangle$, that requires a transfer at $\langle s'_i, l_i\rangle$.

- **Case 2:** Let i be a transportation request going from station $\langle s_i, l_i\rangle$ towards $\langle s'_i, l_i\rangle$. If we assume that i comes from a previous line and has transferred at $\langle s_i, l_i\rangle$, then t_i is the time slot that minimizes the waiting time at $\langle s_i, l_i\rangle$. However, a line run located earlier than t_i cannot serve request i. Thus, the inconvenience costs of this kind of request is maximum before t_i as depicted in Figure 1.3.

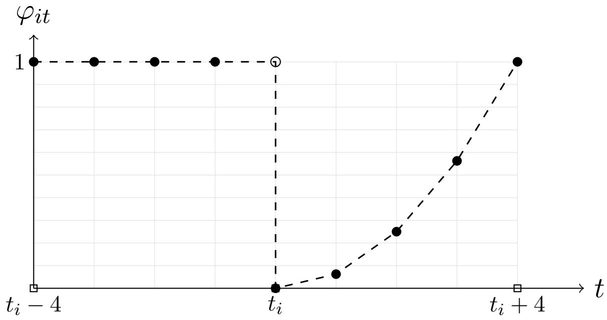


Figure 1.3: Inconvenience costs for a transportation request i, going from station $\langle s_i, l_i \rangle$ towards $\langle s'_i, l_i \rangle$, requiring a transfer at $\langle s_i, l_i \rangle$.

According to the survey conducted by Stern (1996) on various transit agencies in the US, it is infrequent that passengers use more than a single transfer during their origin-destination trips on the regional transit network. In these circumstances, it is reasonable to classify requests into two subsets: requests that use a single line to reach their destinations and those that require to perform a single transfer in order to achieve the final destination. For the first type, the symmetrical function shown in Figure 1.1 can be applied to assess the user's inconvenience and, for the second type of request, the previous asymmetrical functions can provide an extension to the scenario where the presence of transfers can be treated with the methodology developed throughout the chapter. Moreover, the penalization costs associated to the non-served requests can be weighted according to the case where the affected user belongs.

1.7 Conclusions

A new approach for jointly planning timetables and vehicle schedules along a single transit line has been developed by emphasizing the point of view of potential customers. The setting analyzed in this chapter assumes a model of fully disaggregated demand for a scenario that includes capacity constraints and demand behavior according to different criteria. A p-median based formulation has been proposed including specific constraints for the scheduling problem for a given fleet size of vehicles. In addition, demand behavior is associated with the inclusion of closest assignment type constraints.

A clustering algorithm has been developed in order to provide an alternative methodology for solving instances of the problem when computational time must be limited. The performed computational experience shows the difficulty of including closest assignment constraints in a transportation problem and the advantages of deriving a clustering algorithm that allows an appropriate preprocessing of the information.

The infrastructure analyzed in the chapter consists of a single corridor. However, we have developed some hints on how to extend our methodology to more general transportation networks where multiple transit lines operate. In those contexts, transfers between lines required by the passengers can modify the cost associated to the user's inconvenience. The optimization model is shown as a consistent approach, since its applicability remains despite of the change of the network infrastructure.

1.8 Appendix 1: Notation

Data:

L	transit corridor
S	node set (stations)
$e \in E$	edge set (tracks)
$l \in \mathcal{L}$	set of feasible lines in L ($\mathcal{L} = \{1, 2\}$)
$l = 1$	directed transit line running along L
$l = 2$	directed transit line running along L in the opposite direction of $l = 1$
$S_l \subseteq S$	set of stations for a given line $l \in \mathcal{L}$
$\langle s, l\rangle \in S_l$	station in position $s \in \{1, ..., \lvert S_l\rvert\}$
$E_l \subset E$	subset that contains all edges used by line l
$t \in T$	set of time slots ($T = \{1, ..., \lvert T\rvert\}$)
ρ_l	number of line runs to locate
Q	vehicle capacity
κ	fleet size of a line
$i \in I$	set of transportation requests
l_i	line used by request i
$\langle s_i, l_i\rangle$, $\langle s'_i, l_i\rangle$	origin and destination stations for request i
m_{ie}	parameter equal to one if edge $e \in E_{l_i}$ is used when request $i \in I$ is served or equal to zero otherwise
t_i	preferred departure time for request i to locate a line run in station $\langle 1, l_i\rangle$
t_i^-, t_i^+	earliest and latest times that are admissible for serving request i
c_i	penalization cost that will be paid if request i is not served
$T_i \subseteq T$	set of feasible time slots where a line run can be located in order to serve request i ($T_i = \{t_i^-, t_i^- + 1, ..., t_i^+\}$)
φ_{it}	relative cost of allocating request i to the line run which departs from station $\langle 1, l_i\rangle$ at time $t \in T_i$
$I_{lt} \subset I$	subset of requests that can be served locating a line run for line l in time slot t
$r \in R_l$	set of line runs defined in line l ($\lvert R_l\rvert = \rho_l$)
Θ	timetable along partition T
$\theta^+_{\langle s,l\rangle r}, \theta^-_{\langle s,l\rangle r}$	arrival/departure times at station $\langle s, l\rangle$ for line run $r \in R_l$ in line $l \in \mathcal{L}$
$\lambda_{\langle s,l\rangle}$	stopping time required at station $\langle s, l\rangle$
$\mu_{\langle s,l\rangle}$	travel time between stations $\langle s, l\rangle$ and $\langle s + 1, l\rangle$
τ_l	fixed travel time required to complete a line run in line l

Decision Variables:

$x_{lt} \in \{0,1\}$	binary variable equal to 1 if and only if a line run is allocated to line l at time slot t
$y_{it} \in \{0,1\}$	binary variable equal to 1 if and only if request i is allocated to a vehicle which starts a line run at time t

Clustering algorithm data:

$\bar{I}_{lt} \subset I$	subset of requests with a common preferred departure time for locating a line run in line l at the time slot t
$j \in J$	set of clustered requests
q_j	weight of a clustered request
q^{max}	maximum weight for a clustered request
$\bar{t}_j$	preferred departure time for request i to locate a line run in station $\langle 1, l_i \rangle$
$\bar{c}_j$	penalization cost that will be paid if the clustered request j is not served
m_i	edge usage of request i ($m_i = [m_{i1}, m_{i2}, ..., m_{i\|E_{l_i}\|}]$)
$\bar{m}_{je}$	number of requests in I that are grouped in j and use edge e ($\bar{m}_j = [\bar{m}_{j1}, \bar{m}_{j2}, ..., \bar{m}_{j\|E_{l_j}\|}]$)

Bibliography

Brannlund, U., Lindberg, P. O., Nou, A. and Nilsson, J.-E. (1998) "Railway timetabling using Lagrangian relaxation". *Transportation Science*, vol. 32(4): 358–369.

Bunte, S. and Kliewer, N. (2009) "An overview on vehicle scheduling models". *Public Transport*, vol. 1: 299–317.

Cadarso, L. and Marín, A. (2012) "Integration of timetable planning and rolling stock in rapid transit networks". *Annals of Operations Research*, vol. 199(1): 113–135.

Cai, X. and Goh, C. J. (1994) "A fast heuristic for the train scheduling problem". *Computers & Operations Research*, vol. 21(5): 499–510.

Caprara, A., Fischetti, M. and Toth, P. (2002) "Modeling and solving the train timetabling problem". *Operations Research*, vol. 50(5): 851–861.

Caprara, A., Monaci, M., Toth, P. and Guida, P. L. (2006) "A Lagrangian heuristic algorithm for a real-world train timetabling problem". *Discrete Applied Mathematics*, vol. 154(5): 738–753.

Castelli, L., Pesenti, R. and Ukovich, W. (2004) "Scheduling multimodal transportation systems". *European Journal of Operational Research*, vol. 155(3): 603–615.

Ceder, A. and Wilson, N. H. M. (1986) "Bus network design". *Transportation Research Part B: Methodological*, vol. 20(4): 331–344.

Chakroborty, P., Deb, K. and Sharma, R. K. (2001) "Optimal fleet size distribution and scheduling of transit systems using genetic algorithms". *Transportation Planning and Technology*, vol. 24(3): 209–226.

Chang, S.-C. and Chung, Y.-C. (2005) "From timetabling to train regulation–a new train operation model". *Information and Software Technology*, vol. 47(9): 575–585.

Chew, K., Pang, J., Liu, Q., Ou, J. and Teo, C. (2001) "An optimization based approach to the train operator scheduling problem at Singapore MRT". *Annals of Operations Research*, vol. 108(1): 111–118.

Cordeau, J.-F., Laporte, G., Potvin, J.-Y. and Savelsbergh, M. W. (2007a) "Chapter 7: Transportation on Demand". In C. Barnhart and G. Laporte, eds., *Transportation*, vol. 14 of *Handbooks in Operations Research and Management Science*, pp. 429–466. Elsevier.

Cordeau, J.-F., Laporte, G., Savelsbergh, M. W. and Vigo, D. (2007b) "Chapter 6: Vehicle Routing". In C. Barnhart and G. Laporte, eds., *Transportation*, vol. 14 of *Handbooks in Operations Research and Management Science*, pp. 367–428. Elsevier.

de Palma, A. and Lindsey, R. (2001) "Optimal timetables for public transportation". *Transportation Research Part B: Methodological*, vol. 35(8): 789–813.

Espejo, I., Marín, A. and Rodríguez-Chía, A. M. (2012) "Closest assignment constraints in discrete location problems". *European Journal of Operational Research*, vol. 219(1): 49–58.

Fosgerau, M. (2009) "The marginal social cost of headway for a scheduled service". *Transportation Research Part B: Methodological*, vol. 43(8–9): 813–820.

Grosfeld-Nir, A. and Bookbinder, J. (1995) "The planning of headways in urban public transit". *Annals of Operations Research*, vol. 60: 145–160.

Guihaire, V. and Hao, J.-K. (2008) "Transit network design and scheduling: A global review". *Transportation Research Part A: Policy and Practice*, vol. 42(10): 1251–1273.

Guihaire, V. and Hao, J.-K. (2010) "Transit network timetabling and vehicle assignment for regulating authorities". *Computers & Industrial Engineering*, vol. 59(1): 16–23.

Guo, Z. and Wilson, N. H. (2011) "Assessing the cost of transfer inconvenience in public transport systems: A case study of the london underground". *Transportation Research Part A: Policy and Practice*, vol. 45(2): 91–104.

Hakimi, S. (1964) "Optimum locations of switching centers and the absolute centers and medians of a graph". *Operations Research*, vol. 12(3): 450–459.

Hakimi, S. (1965) “Optimum distribution of switching centers in a communications network and some related graph theoretic problems”. *Operations Research*, vol. 13(3): 462–475.

Higgins, A., Kozan, E. and Ferreira, L. (1996) “Optimal scheduling of trains on a single line track”. *Transportation Research Part B: Methodological*, vol. 30(2): 147–161.

Liebchen, C., Schachtebeck, M., Schöbel, A., Stiller, S. and Prigge, A. (2010) “Computing delay resistant railway timetables”. *Computers & Operations Research*, vol. 37(5): 857–868.

Liu, Z.-G. and Shen, J.-S. (2007) “Regional bus operation bi-level programming model integrating timetabling and vehicle scheduling”. *Systems Engineering–Theory & Practice*, vol. 27(11): 135–141.

Magnanti, T. and Wong, R. (1984) “Network design and transportation planning: Models and algorithms.” *Transportation Science*, vol. 18(1): 1–55.

Oliveira, E. and Smith, B. M. (2000) “A job-shop scheduling model for the single-track railway scheduling problem”. Tech. rep., 2000.21, University of Leeds, Leeds, UK.

Quak, C. (2003) *Bus line planning.* Ph.D. thesis, Delft University of Technology. The Netherlands.

Small, K. A. (1982) “The scheduling of consumer activities: Work trips”. *The American Economic Review*, vol. 72(3): 467–479.

Stern, R. (1996) “Passenger transfer system review. synthesis of transit practice”. *Transportation Research Board*, vol. 19.

Wagner, J. L. and Falkson, L. M. (1975) “The optimal nodal location of public facilities with price-sensitive demand”. *Geographical Analysis*, vol. 7(1): 69–83.

Zhou, X. and Zhong, M. (2005) “Bicriteria train scheduling for high-speed passenger railroad planning applications”. *European Journal of Operational Research*, vol. 167(3): 752–771.

Zhou, X. and Zhong, M. (2007) “Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds”. *Transportation Research Part B: Methodological*, vol. 41(3): 320–341.

Chapter 2

Integrating timetables, vehicle schedules and passenger routings in a transit network

ABSTRACT

The Transit Network Timetabling and Scheduling Problem (TNTSP) aims at determining an optimal timetable for each line in a transit network by establishing departure and arrival times at each station and allocating a vehicle to each timetable. The current models for planning of timetables and vehicle schedules use the *a priori* knowledge of passengers' routings. However, the actual route choice of a passenger depends on the timetable. This paper solves the TNTSP in a public transit network by integrating passengers' routings in the model. The proposed formulation guarantees that each user is allocated to the best possible timetable, while satisfying capacity constraints.

Keywords: Timetabling; vehicle scheduling; schedule delay; location-allocation.

2.1 Introduction

The Transit Network Timetabling and Scheduling Problem (TNTSP) aims at determining an optimal timetable for each line in a transit network by establishing departure and arrival times at each station and allocating a vehicle to each timetable. The input data consist of a Public Transportation Network (PTN) made up of a set of stations and links between them, a set of lines, a fleet of capacitated vehicles, a fixed budget for line runs and an origin/destination matriz. The output data consist of a set of arrival and departure times at the stations for the vehicles. Several criteria can be taken into consideration in these problems, for example waiting times, short transfers, fleet size, travel time, load factor, customer utilization and, in general terms, users' and operators' costs. Traditionally, the Transit Network Timetabling Problem (TNTP) has been studied as a preliminary step for the Vehicle Scheduling Problem (VSP), where the TNTP output is an input for the VSP. Unfortunately, this approach leads to a suboptimal solution for the TNTSP. In this chapter we solve the TNTSP by integrating the TNTP and VSP.

The *scheduled delay* (see, e.g., Small, 1982; de Palma and Lindsey, 2001; Mesa et al., 2014) can be defined as the deviation between desired departure and arrival times and actual ones. We assume that while arriving early is involves some wasted time, arriving late typically has more severe consequences. We define a *strategy* as a combination of an itinerary and a potential timetable that a passenger could choose to travel in a PTN. If we first assume uncapacitated vehicles, each passenger may freely choose the strategy that minimizes his scheduled delay. This means that a suboptimal shortest path in the PTN with a convenient timetable may lead to a user's scheduled delay lower than that corresponding to a shortest path with an inconvenient timetable. If we now include vehicle Capacity Constraints (CC) the PTN operator may have to assign specific itineraries to passengers in order to avoid overloading the vehicles. If this option is not available (as it happens in many bus or metro networks where users freely choose their routes) the PTN operator may have to design the timetables and allocate the vehicles in such a way that capacities are respected in the event when all passengers choose their optimal strategy. This gives rise to a leader-follower behaviour: demand allocation depends on the timetables and vice versa. The constraints we impose to guarantee these conditions are called optimal assignment constraints.

Example 2.1- Consider a user who aims to travel from station A to station B (Figure 2.1) departing from A at 8:30 and reaching B at 9:10. There are two itineraries $A \rightarrow B$ and $A \rightarrow C \rightarrow B$, but in fact this user may choose between three strategies: (1) depart from A at 8:10 to arrive at B at 8:50 (2) depart from A at 8:45 to arrive at B at 9:25 and (3) depart from A at 8:25 to arrive at C at 8:40 and from C at 8:45 to arrive at B at 9:15. In option 1 would 20 minutes are wasted for boarding in advance while option 2 leads to 15 minutes of lateness. Option 3 generates five wasted minutes for boarding in advance and five minutes of lateness. If all these strategies are inconvenient for the user, he can choose an alternative mode of transportation.

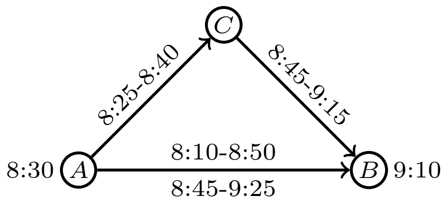


Figure 2.1: Three strategies for traveling from A at 08:30 to B at 9:10.

We define several subproblems of the TNTSP depending on the types of constraints imposed. First, the Vehicle Scheduling Constraints (VSC) impose that the timetables can be operated with the available fleet size. This problem reduces to the Transit Network Timetabling Problem (TNTP) that involves different problems according to how CC are imposed. If there are no CCs, passengers will travel in the PTN by following optimal strategies. This problem is called the User TNTP (TNTP^U). If there are vehicle capacities and the transit operator is able to route passengers in the network, the problem is called Operator TNTP (TNTP^O). However, it is also possible to impose CCs and let passengers choose their path. We call this problem the System TNTP (TNTP^S). Imposing the VSC in the previous subproblems, we obtain respectively the User TNTSP (TNTSP^U), the Operator TNTSP (TNTSP^O) and the System TNTSP (TNTSP^S). It is obvious that the solution spaces of these problems are embedded as illustrated in Figure 2.2.

$$\begin{array}{ccccc} \text{TNTP}^S & \subseteq & \text{TNTP}^O & \subseteq & \text{TNTP}^U \\ \cup\!\shortmid & & \cup\!\shortmid & & \cup\!\shortmid \\ \text{TNTSP}^S & \subseteq & \text{TNTSP}^O & \subseteq & \text{TNTSP}^U \end{array}$$

Figure 2.2: Inclusion relationships between the solution spaces of the TNTSP subproblems.

In this chapter, we study the TNTSP and its variants as described above. After reviewing the main contributions to the TNTSP, we define all items involved in the different subproblems and, in particular, we describe how to compute the available strategies for a transportation request in a PTN. Timetables and vehicle schedules are usually computed assuming the knowledge of passengers' routings from the results of a previous phase. However, the actual route a passenger will take depends on the timetable which is not yet known a priori. In this chapter we integrate passenger route choices within the TNTSP which, as far as we are aware, is a new scientific contribution. This new solution framework is flexible and allows us to optimally allocate transportation requests under vehicle capacity constraints. The six TNTSP subproblems previously defined and their solution spaces are compared. A testbed of randomly generated instances is generated over different network configurations available in the literature and computational results are reported.

The remainder of this chapter is structured as follows. Section 2.2 reviews the most relevant contributions related to this chapter. Section 2.3, provides the description of the problem and the information required to compute itineraries and strategies in a transit network made up of several lines. Section 2.4 presents the mathematical formulation for the integrated TNTSP and for the subproblems just described. Computational results are presented in Section 2.5 followed by conclusions in Section 2.6.

2.2 Background

Our literature review focuses on contributions that integrate the timetabling and the vehicle scheduling problems. For reviews on the TNTP we refer the reader to Cacchiani and Toth (2012); Caprara et al. (2007, 2011), and Lusby et al. (2011). For reviews on the VSP see Törnquist (2007), and Bunte and Kliewer (2009).

New research in transportation planning has focused on the benefits that can be obtained through the integration of different stages belonging to the transit planning process, known as network design, line planning, frequency setting, timetabling, vehicle scheduling and crew rostering (Guihaire and Hao, 2008b). However, not much research has been developed on the integration of timetabling and vehicle scheduling (Ibarra-Rojas et al., 2014). Since the VSP is easy to solve, it is straightforward to implement iterative approaches which modify the current timetable and then, solves the VSP. If a complete integration is desired, the departure times of trips would be decision variables. In this case, network flow formulations and algorithms for the VSP would be difficult to implement since the model lacks a fixed network. Instead, it deals with a set of potential networks that depend on timetabling decisions.

Solution approaches for integrating the two subproblems of the planning process can be divided into two types: sequential integration and complete integration. Sequential integration considers the characteristics of one subproblem while the other one is optimized. However, a sequential approach may lead to suboptimal solutions. We refer to Ceder (2001), van den Heuvel et al. (2008), Guihaire and Hao (2008a), Ceder (2011) and Petersen et al. (2013), for partial integrations related to the TNTSP. On the other hand, a complete integration defines a formulation or a solution methodology to determine the decisions of two or more subproblems simultaneously.

Complete integrations of the timetabling problem with the vehicle scheduling problem have been considered in only a few papers. As far as we know, the first study on a timetabling problem that includes the number of required vehicles in the objective function, is that of Chakroborty et al. (2001). Here, the transit network is made up of multiple lines with a single transfer node. Castelli et al. (2004) deal with non-periodic timetables assuming that routes, means of transport and quality of services are fixed in advance. The operator's main objective is to minimize cost while serving as many users as possible. These authors integrate constraints on the number of available vehicles in the transit network timetabling problem. Liu and Shen (2007), have proposed a bilevel optimization problem consisting of a hierarchical formulation, where one problem (lower level decisions) is embedded within another (upper level decisions). The upper level solves the vehicle scheduling problem with the objective of minimizing the fleet size and the deadheading time, while the lower level optimizes the timetabling decisions in order to minimize the total transfer time for passengers. Fleurent and Lessard (2009) established a measure function for the timetabling problem that incorporates key elements of synchronization, such as the number of passengers transferring from one line to another and the related waiting time. These authors include vehicle scheduling decisions and other measures, such as vehicle usage costs. Guihaire and Hao (2010) integrate the timetabling and vehicle scheduling problems through an optimization model with a weighted objective function that considers the quantity and quality of transfers, the evenness of headway times, the fleet size, and the length of the deadheads. One of the assumptions is the existence of an initial timetable and feasible time intervals for departures and arrivals. This information serves to design line and trip shift movements that are used to modify the feasible timetable and then find the optimal vehicle schedule.

More recently, Ibarra-Rojas et al. (2014) proposed a bi-objective optimization problem to jointly solve the single depot VSP and the bus timetabling problem by considering time windows for departure times and assuming constant demand. The objectives are the maximization of the number of passengers who benefit by well-timed transfers, and the minimization of the fleet size. The authors implemented an ϵ-constraint algorithm to obtain Pareto-optimal solutions. Numerical results show that in some instances using one more vehicle leads to significant reductions in the number of passenger transfers.

Customer-oriented optimization of public transport requires data about the passengers in order to obtain realistic models. Current models take passenger data into account by using the following two-phase approach. In a first phase, routes for the passengers are determined. In a second phase, the actual planning of timetables takes place using the knowledge of which routes passengers wish to travel given the results of the first phase. However, the actual route a passenger will take strongly depends on the timetable, which is not yet known in the first phase. Hence, the two-phase approach often yields non-optimal solutions. Almost all available models assume that passenger routes are fixed before the design of a timetable starts but this topic has recently received more attention (see Siebert and Goerigk, 2013; Schmidt and Schöbel, 2015b,a and references therein).

Combining information on timetables, vehicle scheduling and passenger choices, Mesa et al. (2014) presented an integrated approach for jointly planning timetables and vehicle schedules along a single transit line while emphasizing the point of view of potential users. The setting analyzed assumes a model of fully disaggregated demand for a scenario that includes capacity constraints and demand behavior according to different criteria. These authors propose a p-median-based formulation is proposed that includes specific constraints for the VSP. Demand behavior is handled through the inclusion of closest assignment type constraints. The authors developed a clustering based algorithm in order to provide an alternative methodology for solving instances of the problem when computational time is limited. Their computational experiments show the difficulty of including closest assignment constraints in a transportation problem and the advantages of deriving a clustering algorithm that allows an appropriate preprocessing and handling of the data.

This chapter differs significantly from previous research. First, we develop a framework for integrating the TNTP, the VSP and passenger routings. As far as we now, the integration of these three problems has never previously been studied. Our approach not only pursues transfer coordination but also users' preferences in terms of departure and arrival times for a fully disaggregated demand. Moreover, each transportation request is treated individually, considering hard time windows constraints for trip duration, departure and arrival times, as well as inconvenience costs related to deviations from these times. Second, we formulate the TNTSP starting from the TNTP and adding constraints regarding to capacities, optimal passenger assignment and fleet size. Finally, these formulations are tested and compared on a testbed of random instances and on different networks as similarly proposed in previous studies in the literature.

2.3 Problem description and formulation

We now formally define and formulate the problem. The reader is referred to Appendix A for a full list of the notation used.

2.3.1 Infrastructure

We first distinguish between the infrastructure network and the network containing all lines and walking corridors for transferring between different lines. A *Public Transportation Network* (PTN) is a graph $G = (S, A)$ with a set of nodes S representing stations and a set of arcs A, where each arc represents a direct connection between two stations of S. Given a PTN, $G = (S, A)$, a *Public*

Transportation Line (PTL) is a connected directed graph $l \equiv (S_l, A_l)$ where l belongs to a set of lines $\mathcal{L}$ and is identified with a set of nodes $S_l \subseteq S$, representing stations, and a set of arcs $A_l \in A$, where each arc represents a direct connection between two stations using line l. We distinguish between two types of lines. The set of directed path lines $\overline{\mathcal{L}} \subseteq \mathcal{L}$ and a set of directed cycle lines $\mathring{\mathcal{L}} \subseteq \mathcal{L}$. Moreover, the set of lines $\mathcal{L}$ can be split into the set of lines going forward, $\overrightarrow{\mathcal{L}} \subseteq \mathcal{L}$ and the set of lines going backwards, $\overleftarrow{\mathcal{L}} \subseteq \mathcal{L}$. Therefore, it is obvious that $\mathcal{L} = \overline{\mathcal{L}} \cup \mathring{\mathcal{L}} = \overrightarrow{\mathcal{L}} \cup \overleftarrow{\mathcal{L}}$ and, in addition, each line $l \in \overrightarrow{\mathcal{L}}$ is identified with its corresponding line $l' \in \overleftarrow{\mathcal{L}}$ (running in opposite direction) by means of $l' = l + |\overrightarrow{\mathcal{L}}|$. A *terminal station* is a station where line runs can start and finish, so it can be stated that path lines have two terminal stations (at both ends of the line) and cyclic lines have only one terminal station.

Example 2.2- Figure 2.3 shows a PTN of 4 nodes $S = \{1, 2, 3, 4\}$ and 8 arcs $A = \{(1, 2), (2, 1), (2, 4), (2, 3), (3, 2), (3, 4), (4, 2), (4, 3)\}$.

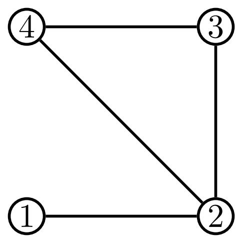


Figure 2.3: A PTN with four nodes (stations) and eight arcs.

In the graph of Figure 2.3 we define lines $\mathcal{L} = \{1, 2, 3, 4\}$ where $A_1 = \{(1, 2), (2, 3)\}, A_2 = \{(2, 3), (3, 4), (4, 2)\}, A_3 = \{(3, 2), (2, 1)\}, A_4 = \{(2, 4), (4, 3), (3, 2)\}$. In addition we can define sets $\overline{\mathcal{L}} = \{1, 3\}$, $\mathring{\mathcal{L}} = \{2, 4\}$, $\overrightarrow{\mathcal{L}} = \{1, 2\}$, $\overleftarrow{\mathcal{L}} = \{3, 4\}$.

□

Given a PTN $G = (S, A)$, the associated *Change&Go Network* (CGN) is a graph $\mathcal{G}$ defined in order to include transfer activities between lines (see Schöbel and Scholl, 2006). It can be denoted as $\mathcal{G} = (\mathcal{N} \cup \ddot{\mathcal{N}}, \mathcal{A} \cup \mathcal{A}^{(tra)})$, where $(l, i) \in \mathcal{N}$ is the set of nodes of each line, $\ddot{\mathcal{N}} = \{|S| + 1, ..., |\ddot{\mathcal{N}}|\}$ is the set of transfer nodes between lines, $(l, i, j) \in \mathcal{A}$ is the set of arcs of all lines $l \in \mathcal{L}$ and $(l, i, n) \in \mathcal{A}^{(tra)}$ is the set of transfer edges between lines.

Basically, the CGN replicates each node of the PTN once for each line and analogously with the arcs. In addition, transfer nodes are added to the CGN as well as transfer edges for each line.

Example 2.3- Given the PTN and the linepool defined in Example 2.2, the associated CGN is depicted in Figure 2.4.

In the graph of Figure 2.4 with corresponding line nodes $\mathcal{N} = \{(1, 1), (3, 1), (1, 2), (2, 2), (3, 2), (4, 2), (1, 3), (2, 3), (3, 3), (4, 3), (2, 4), (4, 4)\}$, transfer nodes $\ddot{\mathcal{N}} = \{5, 6\}$, line arcs $\mathcal{A} = \{(1, 1, 2), (1, 2, 3), (2, 2, 3), (2, 3, 4), (2, 4, 2), (3, 3, 2), (3, 2, 1), (4, 2, 4), (4, 4, 3), (4, 3, 2)\}$ and transfer arcs $\mathcal{A}^{(tra)} = \{(1, 2, 5), (1, 5, 2), (2, 2, 5), (2, 5, 2), ..., (4, 3, 6), (4, 6, 3)\}$.

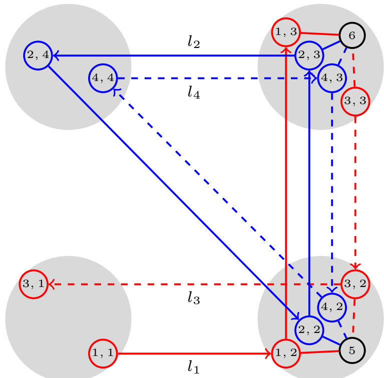


Figure 2.4: Change&Go Network associated to the network of Figure 2.3.

2.3.2 Timetables and vehicle schedules

Given a transit line $l \in \mathcal{L}$, a *line run* is an expedition made up by a vehicle making stops for boards and alights at every station along the line. A discretization T of the continuous time interval is assumed in order to assign departures of the κ available vehicles to the set of time slots $t \in T$. Since all vehicles will have the same circulation speed and stopping time at every (non terminal) station along l, one can assume a fixed travel time τ_l to complete a line run on l. We denote by c_l the cost of implementing a line run in l and a maximum cost ρ is allowed to implement all line runs. All vehicles have the same capacity Q.

A vehicle allocated to $l \in \overrightarrow{\mathcal{L}}$ can start a line run at any time slot $t \in T$. The transit line requires a travel time τ_l (including intermediate stops) to be traversed in one direction. Once the line run is completed, the associated vehicle becomes part of the fleet size of the line $l + |\overrightarrow{\mathcal{L}}|$ ($l - |\overrightarrow{\mathcal{L}}|$ if $l \in \overleftarrow{\mathcal{L}}$), and a new line run can be started in any time slot $t' \in T$ such that $t' \geq t + \tau_l$. Circular lines, are similar to path lines except that it has only one terminal station. Therefore, any itinerary that involves traversing the terminal station will require a transfer at that station.

Example 2.4- (cont) In the graph of Figure 2.4, once a vehicle of line 1 reaches station 3, it changes its direction and becomes a vehicle of line 3. The same occurs for lines 2 and 4 once a terminal station is fixed (for example at station 2). This means that once a vehicle of line 2 reaches station 2, it becomes a vehicle of line 4 (it cannot be used in a line run of line 2 at that moment). Thus, if a passenger travels from station 4 to station 3 using line 2, a transfer will be required at station 2 towards line 2 (again) or line 1.

2.3.3 Demand

Each user has fixed upper and lower bounds associated to the departure and arrival times. Additionally other inconveniences related to in-vehicle times, line-change penalties and deviation between desired departure and arrival times will be taken into account. The concept of *schedule delay*, introduced by Small (1982), arises with the fact that arriving early is likely to involve some time wasted while for most users, arriving late has more severe repercussions. Let I be the set of user transportation requests. In what follows, we use the terms user and request indiscriminately. Each request $i \in I$ involves the following information:

1. An origin and a destination stations.

2. Preferred departure times t_i and $t_{i+|I|}$ to start and finish the trip respectively. Furthermore, t_i^- denotes the earliest time at which user i can start the trip and $t_{i+|I|}^+$ the latest time at which i can reach his destination. Note that, these parameters are dependent on the travel time estimated by the user considering the structure of the PTN (see Example 2.5).

3. A penalty of one unit is paid if request i is not served.

In addition we will introduce in the following section an inconvenience parameter to measure the service quality perceived by each user.

2.3.4 Strategies

Given a CGN, a *hyperpath* is the set of all possible itineraries connecting an origin and a destination. Each itinerary offers different travel *options* for traveling according to each combination of the potential timetables from the different lines that can be used for completing a trip.

Let $\pi \in \Pi$ be the set of all itineraries and $\Pi_i \subseteq \Pi$ the subset of itineraries that can be used by request i. Note that each itinerary π is related with a set of lines $\mathcal{L}_\pi$ and transfer nodes used to complete a trip. Once the path is defined, the user can consider different options of departure times, depending on the combinations of timetables that can be implemented on each line of the path. By $r \in \mathcal{R}_{i\pi}$ we denote the set of options that can be used to serve request i by means of itinerary π. Given this notation, we can define an inconvenience cost function parameter $\varphi_{i\pi r}$ that computes the cost of allocating request i to itinerary π and option r. Additionally, in order to keep track of the capacity usage, we define a binary parameter $m_{\pi a}$ equal to one if and only if arc $a \in A$ is used along itinerary π. In addition, we denote by $t_{i\pi rl}$ the departure time slot used for a vehicle serving request i on line l when itinerary π and option r are used. Note that, in order to simplify the notation, we will consider strategies involving one transfer at most. This makes sense given the number of transfers that users are usually willing to perform in practice (Stern, 1996). However, the proposed setting allows the implementation of several transfers in a general CGN.

Example 2.5- Users can travel inside the network of Figure 2.3 following different strategies. As an example, a passenger ($i = 1$) going from $s = 4$ to $s = 3$ can travel choosing one of the following itineraries:

- itinerary 1, involving a trip from node $(4, 4)$ to node $(4, 3)$ (denoted by $(4, 4) \rightarrow (4, 3)$ in the following),

- itinerary 2, involving two trips $(4, 4) \rightarrow (4, 2)$, $(4, 2) \rightarrow (4, 3)$ and a intermediate transfer (at station 2),

- itinerary 3, involving two trips $(4,4) \rightarrow (4,2)$, $(1,2) \rightarrow (1,3)$ and a intermediate transfer (at station 2).

For simplicity, we assume that distances between adjacent stations of the PTN are all equal to 1. This way a passenger ($i = 2$) traveling from station 1 to station 3 can only choose the itinerary $(1,1) \rightarrow (1,2) \rightarrow (1,3)$ since in this case it does not make sense to transfer at station 2 to another line. The different options for passenger $i = 2$ and his only one available itinerary are (assuming $t_2 = 9$, $t_{2+|I|} = 11$, $t_2^- = 5$ and $t_{2+|I|}^+ = 15$) starting his trip at any of the times $\{5, 6, 7, 8, 9, 10, 11, 12, 13\}$. The different options for passenger $i = 1$ and itinerary 2, are (assuming $t_1 = 9$, $t_{1+|I|} = 10$, $t_1^- = 5$ and $t_{1+|I|}^+ = 14$) the following:

- starting his trip at time 5 and transferring at any of the times $\{6, 7, 8, 9, 10, 11, 12, 13\}$
- starting his trip at time 6 and transferring at any of the times $\{7, 8, 9, 10, 11, 12, 13\}$
- starting his trip at time 7 and transferring at any of the times $\{8, 9, 10, 11, 12, 13\}$
- starting his trip at time 8 and transferring at any of the times $\{9, 10, 11, 12, 13\}$
- starting his trip at time 9 and transferring at any of the times $\{10, 11, 12, 13\}$
- starting his trip at time 10 and transferring at any of the times $\{11, 12, 13\}$
- starting his trip at time 11 and transferring at any of the times $\{12, 13\}$
- starting his trip at time 12 and transferring at time 13.

The scheduled delay costs will depend on the selected combination itinerary-option. See Section 2.5 for an example of the inconvenience cost function $\varphi_{i\pi R}$.

□

2.4 Formulation

In this section we define the six models described in Section 2.2. We make decisions concerning to the number of line runs located on each line (ρ_l) and on the number of vehicles initially available on each line (κ_l). In addition we make binary decisions by means of variable x_{lt} equal to one if a line run is located on line l at time slot t, and by means of variable $y_{i\pi r}$ equal to one if and only if user i is allocated to itinerary $\pi \in \Pi_i$ and option $r \in \mathcal{R}_{i\pi}$.

We consider first the problem of establishing the departure times from terminal stations of the lines. Each departure performs a line run from line l that has an associated cost c_l. The formulation is as follows:

$$\min \sum_{i \in I} [\sum_{\pi \in \Pi_i} \sum_{\mathcal{R}_{i\pi}} \varphi_{i\pi r} y_{i\pi r} + (1 - \sum_{\pi \in \Pi_i} \sum_{\mathcal{R}_{i\pi}} y_{i\pi r})] \quad (2.1a)$$

$$s.t. \quad \sum_{\pi \in \Pi_i} \sum_{\mathcal{R}_{i\pi}} y_{i\pi r} \leq 1 \qquad i \in I \quad (2.1b)$$

$$\sum_{t \in T_l} x_{lt} \leq \rho_l \qquad l \in \mathcal{L} \quad (2.1c)$$

$$|\mathcal{L}_\pi| y_{i\pi r} \leq \sum_{l \in \mathcal{L}_\pi} x_{lt_{i\pi rl}} \qquad i \in I, \pi \in \Pi_i, r \in \mathcal{R}_{i\pi} \quad (2.1d)$$

$$\sum_{l \in \mathcal{L}} c_l \rho_l \leq \rho \quad (2.1e)$$

$$x_{lt} \in \{0, 1\} \qquad l \in \mathcal{L}, t \in T_l \quad (2.1f)$$

$$y_{i\pi r} \in \{0, 1\} \qquad i \in I, \pi \in \Pi_i, r \in \mathcal{R}_{i\pi} \quad (2.1g)$$

The objective function (2.1a) minimizes the total user's inconvenience. The first part computes the inconvenience of the passengers using the system whereas the second part counts the penalty cost of those passengers not assigned to any service of the system. Constraints (2.1b) ensure that no more than one strategy is selected for passenger i, avoiding negative terms in the second part of the objective function. Constraints (2.1c) ensure that no more than ρ_l line runs are located on each line $l \in \mathcal{L}$. Constraints (2.1d) ensure that no request will be allocated to a strategy that does not exist. Constraint (2.1e) ensures that the total cost incurred by the line runs is not exceeded.

We recall that TNTP^U does not take into account the fleet size or the capacity of the vehicles. Constraints (2.2) ensure that the number of requests that use an arc of the CGN in a time slot do not exceed Q.

$$\sum_{i \in I} \sum_{\pi \in \Pi_i : l \in \mathcal{L}_\pi} \sum_{r \in \mathcal{R}_{i\pi} : t_{i\pi rl} = t} y_{i\pi r} m_{\pi a} \leq Q x_{lt} \qquad l \in \mathcal{L}, a \in A_l, t \in T \quad (2.2)$$

The TNTP^U together with constraint (2.2) defines the TNTP^O.

Assuming that users know in advance all information related to the itineraries and timetables, it is reasonable to assume that they will choose the strategy that enables them to reach their destination at minimum inconvenience cost. In addition, as previously mentioned, the network manager may not be able to route transportation requests within the CGN. Therefore, in order to model the TNTP^S, two additional constraints can be added. First, we impose that a passenger can be allocated to a line run only if there is at least one strategy available for i:

$$\sum_{l \in \mathcal{L}_\pi} x_{lt_{i\pi rl}} \leq (|\mathcal{L}_\pi| - 1) + \sum_{\pi' \in \Pi_i} \sum_{r' \in \mathcal{R}_{i\pi}} y_{i\pi' r'} \qquad i \in I, \pi \in \Pi_i, r \in \mathcal{R}_{i\pi}. \quad (2.3)$$

Second, constraint (2.4) ensures that each request is allocated to a line run if and only if such line run is the one with lowest inconvenience cost:

$$\sum_{l\in\mathcal{L}_\pi} x_{lt_{i\pi rl}} + \sum_{\pi'\in\Pi_i} \sum_{\substack{r'\in\mathcal{R}_{i\pi}:\\ \varphi_{i\pi'r'}>\varphi_{i\pi r}}} y_{i\pi'r'} \leq |\mathcal{L}_\pi| \qquad i\in I, \pi\in\Pi_i, r\in\mathcal{R}_{i\pi}. \tag{2.4}$$

Constraint (2.4) is adapted from Wagner and Falkson (1975) and belongs to the so-called closest assignment constraints. According to Espejo et al. (2012), this kind of constraints can be modelled in many different ways, giving rise to better or worse linear programming relaxations. As shown in that paper, the constraint provided by Wagner and Falkson (1975) can be strengthened by using the fixed numbers of facilities (line runs) to locate. Even with such improvements, the inclusion of constraints (2.3) and (2.4) considerably increases the computational complexity of the proposed model.

In addition to the previous considerations we can assume the the timetables are carried out by a limited number κ of vehicles. On each line l, vehicles change the direction of the line once the terminal station is reached and then become a vehicle available for line $l+|\overrightarrow{\mathcal{L}}|$ ($l-|\overrightarrow{\mathcal{L}}|$ if $l\in\overleftarrow{\mathcal{L}}$). We describe the vehicle scheduling problem through the following set of constraints:

$$\sum_{t'=1}^{\tau_l} x_{lt'} \leq \kappa_l \qquad l\in\mathcal{L} \tag{2.5a}$$

$$\sum_{t'=1}^{t} x_{lt'} - \sum_{t'=1}^{t-\tau_l} x_{l+|\overrightarrow{\mathcal{L}}|,t'} \leq \kappa_l \qquad l\in\overrightarrow{\mathcal{L}}, t\in T: t>\tau_l \wedge t<|T|-\tau_l \tag{2.5b}$$

$$\sum_{t'=1}^{t} x_{lt'} - \sum_{t'=1}^{t-\tau_l} x_{l-|\overrightarrow{\mathcal{L}}|,t'} \leq \kappa_l \qquad l\in\overleftarrow{\mathcal{L}}, t\in T: t>\tau_l \wedge t<|T|-\tau_l \tag{2.5c}$$

$$\sum_{l\in\mathcal{L}} \kappa_l \leq \kappa \tag{2.5d}$$

Constraints (2.5a) ensure that no more than κ_l vehicles depart from each line before any vehicle from the opposite direction has arrived. Constraints (2.5b) and (2.5c) ensure that the difference between the number of vehicles that have departed from a line and the number of vehicles that have arrived to that line (coming from the corresponding line in opposite direction) never exceeds the sum of the fleet sizes corresponding to both lines. Constraint (2.5d) ensures that the fleet size is not exceeded.

The TNTP^U, TNTP^O, TNTP^S together with constraints (2.5a)–(2.5d) are defined as the TNTSP^U, TNTSP^O, TNTSP^S respectively.

2.5 Computational experience

In order to show the applicability of the previous models, we have considered six different networks (Figure 2.5) inspired from some already existing in the literature (see Laporte et al., 1994; Laporte et al., 1997). All networks contain 13 nodes and have similar number of edges (ranging from 12 to 16). Each configuration can be obtained by setting out three lines, obtaining from one to five intersections.

Note that each line, represented with continuous dashed and dotted lines, contains two path lines, except for the cartwheel configuration where the dotted line contains two circular lies.

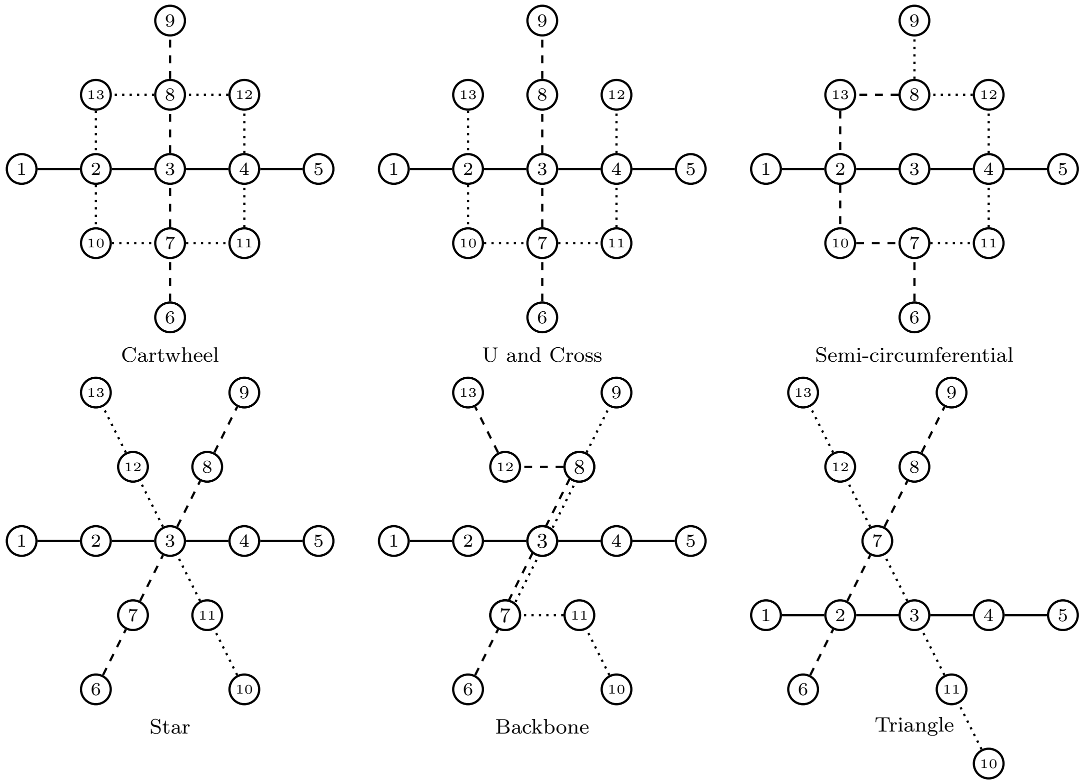


Figure 2.5: Basic configurations obtained from 3 lines.

First, for each O/D pair we have precomputed the different itineraries by using a k-shortest path algorithm (Shier, 1979). We have then generated 10 random instances of transportation requests for $|I| = 100$ with an origin, a destination and a desired arrival time ($t_{i+|I|}$) in $|T| = 60$ (all random values following a discrete uniform distribution). Next, for each configuration, we have computed the desired departure time t_i of each request by means of the corresponding shortest path through the PTN. Each desired departure time t_i lies within a time window $[\max\{0, t_i - 4\}, \min\{|T|, t_i + 4\}]$, or equivalently within a set of feasible time slots $T_i = \{\max\{0, t_i - 4\}, ..., \max\{0, t_i - 1\}, t_i, \min\{|T|, t_i + 1\}, ..., \min\{|T|, t_i + 4\}\}$ (analogously for the desired arrival time $t_{i+|I|}$). The different travel options for traveling were calculated for each user, according to the available itineraries, time windows, and travel times in the network. Each of these options gives rise to a pair $(t_i^*(r, \pi), t_{i+|I|}^*(r, \pi))$ representing actual departure an arrival times for user i using itinerary π and option r. Finally, the inconvenience cost function ($\varphi_{i\pi r}$) was computed using the function

$$\varphi_{i\pi r} = \min\left\{1, \tilde{\varphi}(-|t_i - t_i^*(\pi, r)|^+) + \tilde{\varphi}(|t_{i+|I|}^*(\pi, r) - t_{i+|I|}|^+)\right\}, \tag{2.6}$$

where $|z|^+ = \max\{0, z\}$ and $\tilde{\varphi}$ is a discrete function defined as in Figure 2.6.

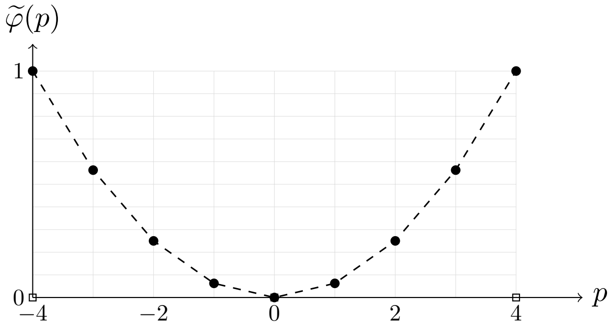


Figure 2.6: Discrete inconvenience costs under consideration.

Line runs vary as $\rho \in \{6, 12, 24\}$ and fleet sizes are defined as $\kappa \in \{6, 12, 24\}$, ensuring that $\kappa \leq \rho$ for each instance. All instances were solved with the MIP Xpress 7.5 optimizer, under a Windows 7 environment in an Intel(R) Core(TM)i7 CPU 2.93 GHz processor and 8 GB RAM. Default values were initially used for all parameters of Xpress solver and a CPU time limit of 3600 seconds was set.

Each of our tables reports the following items. Each row corresponds to a group of 10 instances with the same characteristics $(\#G, |I|, \rho, \kappa, Q)$ indicated in the first five columns. Column t reports the average running time in seconds for the 10 instances corresponding to each row. Column *gapLR* reports the relative gap computed with the best solution found by the solver and the optimal value of the linear relaxation at the root node. Column *nod* indicates the average number of nodes explored in the branch-and-bound tree. Finally, *obj* reports the mean objectives values that we recall represent the total user inconvenience. The reader may note that we do not report gaps because all instances were solved to optimality within the time limit. All tables report analogous information for the different formulations described along the chapter. In order to facilitate the comparison among all tables, we have marked in bold red the best result among all in the same group. In case of ties the best results have been marked in bold blue. Note that, since we are rounding up to 2 decimals, cero values could represent any value in $[0, 0.005]$.

There is dependance between the network configuration and results presented in the tables. First, regarding to the running times, the instance that was more difficult to solve was the Cartwheel (#1) with a significant difference with respect to the other configurations. The main reason for this is because it is the network that offers more possible itineraries and, therefore, strategies. On the contrary, configuration Semi-circumferential (#3) was the fastest to solve since it seems to be the configuration with the poorest directness among the others. Surprisingly, the Backbone (#4) was faster to solve than the Star configuration (#3). In spite of offering #4 more possibilities for completing the trips than #3, this trips are along common-lines over edges (3,7) and (3,8), and this seems to make easier the allocation of passengers. Regarding to U and Cross (#2) and Triangle (#6) we observe a similar performance except for the the TNTSP^U problem where #2 seems to be more complex. With respect to column *gapLR* we observe a direct relationship with column t, that is, those configurations who left a bigger integrality gap at the root node, were those more difficult to solve. Column *nodes* show that, in spite of the *gapLR* values, many of the TNTP^U and TNTSP^U instances could be solved in the root node of the branch and bound tree search just by means of preprocessing

				$TNTP^U$				$TNTP^O$				$TNTP^S$			
#G	$\lvert I\rvert$	ρ	Q	t	$gapLR$	nod	obj	t	$gapLR$	nod	obj	t	$gapLR$	nod	obj
1	100	6	1	7.7	37.86	15	77.2	0.5	0	1	86.6	13.1	2.36	15	90.1
1	100	6	2	7.6	37.86	15	77.2	0.7	0.29	1	81.4	17	1.05	10	82.3
1	100	6	3	7.6	37.86	15	77.2	5.1	1.56	1	79.1	20.4	2.13	26	79.7
1	100	12	1	20.7	49.74	158	61	0.5	0.02	1	76.8	12.8	3.77	19	81.5
1	100	12	2	20.6	49.74	158	61	5.3	1.58	10	68.2	29.2	3.38	165	69.8
1	100	12	3	20.5	49.74	158	61	12.6	5.3	27	63.6	95.9	6.16	1084	64.3
1	100	24	1	18.4	61.99	70	37.6	1.1	0.04	1	62.3	19.3	5.24	71	67.1
1	100	24	2	18	61.99	70	37.6	10.9	5.44	34	46.2	114.4	8.18	4023	48.1
1	100	24	3	18.2	61.99	70	37.6	24.8	21.51	308	40.6	158.2	22.88	1747	41.3
2	100	6	1	2	20.67	1	80.4	0.3	0	1	88.2	2.2	0.64	1	89.7
2	100	6	2	1.9	20.67	1	80.4	0.3	0.09	1	83.7	6.1	0.94	1	84.7
2	100	6	3	1.9	20.67	1	80.4	0.5	0.9	1	81.4	3.6	1.65	1	82.1
2	100	12	1	3.4	33.13	1	66.3	0.3	0.08	1	79.9	2.5	1.92	1	82.3
2	100	12	2	3.4	33.13	1	66.3	0.7	0.79	1	72.2	4.7	1.53	1	73.1
2	100	12	3	3.5	33.13	1	66.3	4.1	3.56	1	68.5	6.8	4.35	1	69.1
2	100	24	1	3.2	47.48	1	45.2	0.3	0	1	67.3	1.1	2.51	1	69.6
2	100	24	2	3.3	47.48	1	45.2	6.3	4.22	154	53.2	17.5	6.01	91	54.4
2	100	24	3	3.2	47.48	1	45.2	8	14.56	112	48.2	26.8	15.97	566	49
3	100	6	1	0	0	1	83.4	0	0	1	90	0	1.14	1	91.7
3	100	6	2	0	0	1	83.4	0	0	1	86.1	0	0.27	1	86.5
3	100	6	3	0	0	1	83.4	0	0.04	1	84.1	0.1	0.34	1	84.4
3	100	12	1	0	0	1	72	0	0	1	83.6	0	1.83	1	85.6
3	100	12	2	0	0	1	72	0	0.05	1	75.1	0.2	0.9	1	75.8
3	100	12	3	0	0	1	72	0	0.03	1	72.4	0.1	0.19	1	72.5
3	100	24	1	0	0	1	57.2	0	0	1	71.6	0	2.14	1	73.6
3	100	24	2	0	0	1	57.2	0	1	1	59.1	0	0.97	1	59.1
3	100	24	3	0	0	1	57.2	0	0.2	1	57.4	0	0.31	1	57.5
4	100	6	1	3.4	22.19	10	85.4	0.2	0	1	90.9	0.6	0.56	1	91.9
4	100	6	2	3.4	22.19	10	85.4	0.6	0.66	1	87.7	0.8	0.89	1	87.9
4	100	6	3	3.5	22.19	10	85.4	1.6	2.34	1	85.9	3	2.34	1	85.9
4	100	12	1	0.9	35.17	1	71.9	0.2	0	1	84.1	1.5	0.77	1	85.4
4	100	12	2	0.9	35.17	1	71.9	1.4	1.58	1	77	8.3	2.19	1	77.6
4	100	12	3	0.9	35.17	1	71.9	0.9	5.5	1	73.7	1.4	5.53	1	73.7
4	100	24	1	1.1	50.92	1	49	0.2	0	1	72.1	0.7	1.1	1	73.3
4	100	24	2	1.1	50.92	1	49	4.6	5.24	1	59	15.4	6.78	893	60.1
4	100	24	3	1.1	50.92	1	49	4.8	13.74	9	52.2	12.7	14.85	51	53
5	100	6	1	0.1	4.19	1	86	0	0.05	1	91.2	0.1	0.57	1	91.9
5	100	6	2	0.1	4.19	1	86	0	0	1	87	0.2	0.02	1	87.1
5	100	6	3	0.1	4.19	1	86	0.1	0.24	1	86	0.2	0.24	1	86
5	100	12	1	0.1	6.28	1	75.6	0	0	1	84.6	0.1	0.44	1	85
5	100	12	2	0.1	6.28	1	75.6	0.1	0.1	1	76.8	0.2	0.48	1	77.2
5	100	12	3	0.1	6.28	1	75.6	0.1	0.93	1	75.6	0.2	0.93	1	75.6
5	100	24	1	0.1	9.23	1	61.4	0	0	1	72.6	0.1	0.41	1	72.9
5	100	24	2	0.1	9.23	1	61.4	0.1	0.82	1	62.2	0.3	1.1	1	62.5
5	100	24	3	0.1	9.23	1	61.4	0.1	2.71	1	61.4	0.2	2.7	1	61.4
6	100	6	1	2.5	20.8	1	83.7	0.2	0.02	1	89.2	1.8	1.66	1	91.2
6	100	6	2	2.4	20.8	1	83.7	0.3	0.4	1	85.5	2.5	0.51	1	85.7
6	100	6	3	2.4	20.8	1	83.7	0.4	2.17	1	83.7	1.1	2.04	1	83.7
6	100	12	1	4.4	33.94	4	70.4	0.2	0	1	82.5	2.4	2.06	1	84.7
6	100	12	2	4.5	33.94	4	70.4	1.7	1.01	1	74.5	6.7	1.64	1	75.3
6	100	12	3	4.3	33.94	4	70.4	3.9	4.85	1	71.1	6.1	4.87	1	71.3
6	100	24	1	5.1	50.12	11	47.6	0.2	0	1	70.4	1.7	2.55	1	72.7
6	100	24	2	5.2	50.12	11	47.6	3.2	4.47	1	55.9	10.9	6.53	203	57.5
6	100	24	3	5.2	50.12	11	47.6	6.4	14.43	18	49.9	13.4	15.12	65	50.5

cuts and heuristics. This could also be achieved in the other problems in some of the instances with a low *gapLR* value. The last column *obj* shows that the lower level of inconvenience is the one provided by configuration #1 (as expected) followed by #2, #6, #3 and #4 (almost the same), and #6. The effect of the common lines in #5, seems to be very convenient for travelers in the among the common segments since the have double possibilities for travelling. However, for those transportation request out from that sector, the common lines only imply a lack of supply.

					$TNTSP^U$				$TNTSP^O$				$TNTSP^S$			
$\#G$	$\lvert I\rvert$	ρ	κ	Q	t	$gapLR$	nod	obj	t	$gapLR$	nod	obj	t	$gapLR$	nod	obj
1	100	12	6	1	32.9	50.16	487	**61.5**	**1.3**	**0.16**	**1**	77.1	24.5	4.99	129	82.7
1	100	12	6	2	32.9	50.16	487	**61.5**	**6.3**	**1.54**	**4**	68.2	45.9	3.49	415	70
1	100	12	6	3	32.8	50.16	487	**61.5**	**12.6**	**5.3**	**16**	63.6	92.3	6.16	567	64.3
1	100	24	6	1	428.1	62.08	25223	**40.2**	**5.4**	**0.71**	**67**	64	176.4	10.07	7102	72
1	100	24	6	2	428.7	62.08	25223	**40.2**	**33**	**7.27**	**251**	48.2	2147.1	11.65	24412	50.9
1	100	24	6	3	**428.8**	62.08	25223	**40.2**	636.4	**24.32**	**16521**	42.8	2281.5	26.45	17749	44
1	100	24	12	1	26.8	61.98	164	**37.6**	**1**	**0.04**	**1**	62.3	20.1	5.27	178	67.1
1	100	24	12	2	26.6	61.98	164	**37.6**	**14.5**	**5.44**	**57**	46.2	131.1	8.18	3166	48.1
1	100	24	12	3	**26.7**	61.98	**164**	**37.6**	44.9	**21.51**	827	40.6	170.9	22.88	2445	41.3
2	100	12	6	1	4.5	33.39	**1**	**66.7**	**0.5**	**0.1**	**1**	80.1	3.8	1.62	**1**	82.5
2	100	12	6	2	4.4	33.39	**1**	**66.7**	**0.7**	**0.5**	**1**	72.4	10.6	2.16	69	74
2	100	12	6	3	4.4	33.39	**1**	**66.7**	**2.6**	**3.31**	**1**	68.6	11.2	4.58	45	69.6
2	100	24	6	1	7.9	47.11	15	**46.3**	**1.9**	**0.21**	**1**	68	15.3	4.05	1556	72
2	100	24	6	2	**7.9**	47.11	**15**	**46.3**	9	**4.51**	166	55	118.2	7.94	4037	57.5
2	100	24	6	3	**7.9**	47.11	**15**	**46.3**	20	**14.87**	158	49.4	132.5	17.41	3378	51
2	100	24	12	1	3.7	47.48	**1**	**45.2**	**0.3**	**0**	**1**	67.3	3.1	2.49	**1**	69.7
2	100	24	12	2	**3.7**	47.48	**1**	**45.2**	7.3	**4.22**	24	53.2	29.2	6.35	1001	54.6
2	100	24	12	3	**3.7**	47.48	**1**	**45.2**	8.7	**14.56**	82	48.2	29.2	15.97	566	49
3	100	12	6	1	**0.1**	**0**	**1**	**72**	**0.1**	**0**	**1**	83.6	**0.1**	1.83	**1**	85.6
3	100	12	6	2	**0.1**	**0**	**1**	**72**	**0.1**	0.05	**1**	75.1	1.4	1.25	**1**	76.1
3	100	12	6	3	**0.1**	**0**	**1**	**72**	**0.1**	0.03	**1**	72.4	**0.1**	0.19	**1**	72.5
3	100	24	6	1	**0.1**	**0**	**1**	**57.8**	0.2	0.05	**1**	71.6	1.7	4.08	59	75.2
3	100	24	6	2	**0.1**	**0**	**1**	**57.8**	**0.1**	0.27	**1**	60.2	3.1	2.06	36	61.5
3	100	24	6	3	**0.1**	**0**	**1**	**57.8**	**0.1**	0.16	**1**	58.2	0.2	0.52	**1**	58.4
3	100	24	12	1	**0.1**	**0**	**1**	**57.2**	**0.1**	**0**	**1**	71.6	**0.1**	2.14	**1**	73.6
3	100	24	12	2	**0.1**	**0**	**1**	**57.2**	0.2	1	**1**	59.1	**0.1**	0.97	**1**	59.1
3	100	24	12	3	**0.1**	**0**	**1**	**57.2**	**0.1**	0.2	**1**	57.4	**0.1**	0.31	**1**	57.5
4	100	12	6	1	3.3	35.24	2	**72**	**0.3**	**0**	**1**	84.1	2.5	0.89	**1**	85.5
4	100	12	6	2	3.3	35.24	2	**72**	**1.3**	**1.56**	**1**	77	9.9	2.17	**1**	77.6
4	100	12	6	3	3.3	35.24	2	**72**	**1.6**	**5.5**	**1**	73.7	2.1	5.52	**1**	73.7
4	100	24	6	1	2.9	50.4	**1**	**49.5**	**1.1**	**0.13**	**1**	72.3	7.5	2.5	329	74.8
4	100	24	6	2	**3**	50.4	**1**	**49.5**	7.2	**5.82**	20	60	251.6	8.21	19754	62.1
4	100	24	6	3	**3**	50.4	**1**	**49.5**	11.7	**15.91**	189	54.2	86.5	17.74	5034	55.5
4	100	24	12	1	1.8	50.92	**1**	**49**	**0.3**	**0**	**1**	72.1	2.3	1.19	**1**	73.3
4	100	24	12	2	**1.8**	50.92	**1**	**49**	4.4	**5.24**	18	59	14.1	6.78	407	60.1
4	100	24	12	3	**1.8**	50.92	**1**	**49**	6.1	**13.74**	33	52.2	12	14.85	58	53
5	100	12	6	1	0.5	6.4	**1**	**75.7**	**0.3**	**0.1**	**1**	84.7	0.7	0.62	**1**	85.3
5	100	12	6	2	0.5	6.4	**1**	**75.7**	**0.2**	**0.19**	**1**	76.9	0.4	0.57	**1**	77.3
5	100	12	6	3	0.5	6.4	**1**	**75.7**	**0.3**	1.06	**1**	**75.7**	0.4	**1.05**	**1**	**75.7**
5	100	24	6	1	0.9	9.01	**1**	**61.8**	**0.1**	**0**	**1**	72.6	0.8	0.58	**1**	73.1
5	100	24	6	2	0.9	9.01	**1**	**61.8**	**0.3**	**0.79**	**1**	63	1	1.51	**1**	63.5
5	100	24	6	3	0.9	9.01	**1**	**61.8**	**0.7**	2.96	**1**	62	1.2	**2.93**	**1**	62
5	100	24	12	1	0.5	9.23	**1**	**61.4**	**0.1**	**0**	**1**	72.6	0.3	0.41	**1**	72.9
5	100	24	12	2	0.5	9.23	**1**	**61.4**	**0.2**	**0.82**	**1**	62.2	0.5	1.1	**1**	62.5
5	100	24	12	3	0.5	9.23	**1**	**61.4**	**0.3**	2.71	**1**	**61.4**	0.5	**2.7**	**1**	**61.4**
6	100	12	6	1	5.7	33.94	**1**	**70.4**	**0.3**	**0**	**1**	82.5	4.7	2.29	**1**	84.9
6	100	12	6	2	5.8	33.94	**1**	**70.4**	**0.7**	**1**	**1**	74.5	8.9	1.6	**1**	75.3
6	100	12	6	3	5.7	33.94	**1**	**70.4**	**2.3**	**4.85**	**1**	71.1	7.2	4.87	4	71.3
6	100	24	6	1	8.4	50.05	39	**49**	**0.5**	**0**	**1**	70.4	7.6	4.19	327	74.1
6	100	24	6	2	8.4	50.05	39	**49**	**7.8**	**5.61**	**21**	57.2	31	7.07	493	58.5
6	100	24	6	3	**8.5**	50.05	**39**	**49**	10.6	**15.38**	86	51.3	60.2	17.06	2424	52.5
6	100	24	12	1	7.3	50.12	**1**	**47.6**	**0.3**	**0**	**1**	70.4	4.3	2.6	**1**	72.8
6	100	24	12	2	7.2	50.12	**1**	**47.6**	**3.4**	**4.47**	**1**	55.9	11.8	6.53	130	57.5
6	100	24	12	3	7.2	50.12	**1**	**47.6**	**4.9**	**14.43**	14	49.9	15	15.12	78	50.5

Comparing the discussed results among the different problems we found that, on these instances, TNTP^O and TNTSP^O were easier solvable than TNTP^U and TNTSP^U respectively in terms of t, $gapLR$ and nod. This makes sense considering that the capacity constraints, reduce significantly the solution space and, therefore, the search space. Those transportation requests that cannot fit in the available vehicles are simply neglected what is not a big difficulty in the allocation problem. This is not the case of the TNTP^S or TNTSP^S that meets a more difficulty in the assignment that must be optimal for each transportation request once that a timetable is fixed (in the x variables). In general terms, adding the Vehicle Scheduling Constraints (VSC) makes more difficult to solve each timetabling variant but instances with $\kappa = 6$ and $\kappa = 12$ does not vary much since a fleet size of $\kappa = 6$ seems to be almost enough to implement $\rho = 24$ line runs.

2.6 Conclusions

In this chapter we have presented a new approach for solving the integration of the Transit Network Timetabling and Scheduling Problem together with the passengers' routing problem. Traditionally, these problems have been studied sequentially but this approach leads to sub-optimal solutions for the entire problem. We present a flexible framework that let us allocate transportation requests to their optimal strategies under capacity constraints. This approach not only pursues transfer coordination but also customers' preferences in terms of preferred departure/arrival times for a fully disaggregated demand. Even more, each transportation request is faced individually, stating hard time windows constraints for departure/arrival times as well as inconvenience costs related to trip duration and time deviations from desired departure/arrival times. A testbed of randomly generated instances has been generated for different network configurations existant in the literature and computational results have been obtained and analyzed.

2.7 Appendix: Notation

Infraestructure:

G	graph corresponding to the PTN
S	node set (stations)
A	set of arcs (indexed by a)
$\mathcal{L}$	set of lines (indexed by l)
$S_l \subseteq S$	subset of stations used by line l
$A_l \subset A$	subset that contains all edges used by line l
$\overline{\mathcal{L}} \subseteq \mathcal{L}$	set of directed path lines
$\mathring{\mathcal{L}} \subseteq \mathcal{L}$	set of directed cycle lines
$\overrightarrow{\mathcal{L}} \subseteq \mathcal{L}$	set of path lines going forward
$\overleftarrow{\mathcal{L}} \subseteq \mathcal{L}$	set of path lines going backwards (a line $l \in \overrightarrow{\mathcal{L}}$ is associated to its respective (opposite) line $l' \in \overleftarrow{\mathcal{L}}$ by means of $l' = l + \lvert\overrightarrow{\mathcal{L}}\rvert$

$(l,i) \in \mathcal{N}$	set of nodes of all lines
$(l,i,j) \in \mathcal{A}$	set of arcs of all lines
$n \in \ddot{\mathcal{N}}$	set of transfer nodes
$\mathcal{A}^{(tra)}$	set of transfer edges

Timetables and vehicle scheduling:

$t \in T$	set of time slots ($T = \{1, ..., \lvert T \rvert\}$)
τ_l	fixed travel time required to complete a line run in line l
Q	vehicle capacity
c_l	cost associated to locate a line run in line l
ρ	total available budget to locate line runs
κ	fleet size

Demand:

$i \in I$	set of transportation requests
t_i, $t_{i+\lvert I \rvert}$	preferred departure and arrival times for request i
t_i^-, t_i^+	earliest and latest times that are admissible for serving request i

Strategies:

$\pi \in \Pi$	set of possible itineraries within the PTN
$\Pi_i \subset \Pi$	subset of itineraries that are valid for user i
$\mathcal{L}_{i\pi} \subseteq \mathcal{L}$	set of lines used by request i when itinerary $\pi \in \Pi_i$ is selected
$r \in \mathcal{R}_{i\pi}$	set of options available for request i when using itinerary $\pi \in \Pi_i$
$\varphi_{i\pi r}$	cost of allocating request i to itinerary π and option option r
$m_{a\pi}$	binary parameter equal to one if itinerary $\pi \in \Pi_i$ occupies arc $a \in A_l$
$t_{i\pi rl}$	time slot that is used for a vehicle departure in line l when the itinerary π and the option r are used

Decision Variables:

ρ_l	number of line runs to locate on line l
κ_l	fleet size to assign on line l
$x_{lt} \in \{0, 1\}$	binary variable equal to 1 when a vehicle starts a line run in line l at time t
$y_{i\pi r} \in \{0, 1\}$	binary variable equal to 1 iif request i is allocated to itinerary $\pi \in \Pi_i$ and option $r \in \mathcal{R}_{i\pi}$

Bibliography

Bunte, S. and Kliewer, N. (2009) “An overview on vehicle scheduling models”. *Public Transport*, vol. 1: 299–317.

Cacchiani, V. and Toth, P. (2012) “Nominal and robust train timetabling problems”. *European Journal of Operational Research*, vol. 219(3): 727–737. Feature Clusters.

Caprara, A., Kroon, L., Monaci, M., Peeters, M. and Toth, P. (2007) *Passenger railway optimization.*, vol. 14, pp. 129–187. Elsevier, Amsterdam.

Caprara, A., Kroon, L. and Toth, P. (2011) *Optimization problems in passenger railway systems*, vol. 6, pp. 3896–3905. Wiley.

Castelli, L., Pesenti, R. and Ukovich, W. (2004) “Scheduling multimodal transportation systems”. *European Journal of Operational Research*, vol. 155(3): 603–615.

Ceder, A. (2001) *Efficient timetabling and vehicle scheduling for public transport*, vol. 505, chap. Computer-Aided Scheduling of Public Transport, pp. 37–52. Springer, Berlin Heidelberg.

Ceder, A. (2011) “Optimal multi-vehicle type transit timetabling and vehicle scheduling”. *Procedia-Social and Behavioral Sciences*, vol. 20: 19–30.

Chakroborty, P., Deb, K. and Sharma, R. K. (2001) “Optimal fleet size distribution and scheduling of transit systems using genetic algorithms”. *Transportation Planning and Technology*, vol. 24(3): 209–226.

de Palma, A. and Lindsey, R. (2001) “Optimal timetables for public transportation”. *Transportation Research Part B: Methodological*, vol. 35(8): 789–813.

Espejo, I., Marín, A. and Rodríguez-Chía, A. M. (2012) “Closest assignment constraints in discrete location problems”. *European Journal of Operational Research*, vol. 219(1): 49–58.

Fleurent, C. and Lessard, R. (2009) “Integrated timetabling and vehicle scheduling in practice”. Tech. rep., GIRO Inc. Montreal, Canada.

Guihaire, V. and Hao, J. (2008a) *Transit network re-timetabling and vehicle scheduling*, vol. 14, pp. 135–144. Springer, Berlin, Heidelberg.

Guihaire, V. and Hao, J.-K. (2008b) “Transit network design and scheduling: A global review”. *Transportation Research Part A: Policy and Practice*, vol. 42(10): 1251–1273.

Guihaire, V. and Hao, J.-K. (2010) “Transit network timetabling and vehicle assignment for regulating authorities”. *Computers & Industrial Engineering*, vol. 59(1): 16–23.

Ibarra-Rojas, O. J., Giesen, R. and Rios-Solis, Y. A. (2014) “An integrated approach for timetabling and vehicle scheduling problems to analyze the trade-off between level of service and operating costs of transit networks”. *Transportation Research Part B: Methodological*, vol. 70(0): 35–46.

Laporte, G., Mesa, J. and Ortega, F. (1994) “Assessing topological configurations for rapid transit networks”. *Stud Locat Anal*, vol. 7: 105–121.

Laporte, G., Mesa, J. and Ortega, F. (1997) “Assesing the efficiency of rapid transit configurations”. *Top*, vol. 5(1): 95–104.

Liu, Z.-G. and Shen, J.-S. (2007) “Regional bus operation bi-level programming model integrating timetabling and vehicle scheduling”. *Systems Engineering–Theory & Practice*, vol. 27(11): 135–141.

Lusby, R., Larsen, J., Ehrgott, M. and Ryan, D. (2011) “Railway track allocation: models and methods”. *OR Spectrum*, vol. 33 (4): 843–883.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2014) “Locating optimal timetables and vehicle schedules in a transit line”. *Annals of Operations Research*, vol. 222: 439–455.

Petersen, H., Larsen, A., Madsen, O., Petersen, B. and Ropke, S. (2013) “The simultaneous vehicle scheduling and passenger service problem”. *Transportation Science*, vol. 47: 603–616.

Schöbel, A. and Scholl, S. (2006) “Line planning with minimal traveling time”. In *Dagstuhl Seminar Proceedings*, p. .

Schmidt, M. and Schöbel, A. (2015a) "The complexity of integrating passenger routing decisions in public transportation models". *Networks*, vol. 65(3): 228–243.

Schmidt, M. and Schöbel, A. (2015b) "Timetabling with passenger routing". *OR Spectrum*, vol. 37(1): 75–97.

Shier, R. D. (1979) "On algorithms from finding the k shortest paths in a network". *Networks*, vol. 9: 195–214.

Siebert, M. and Goerigk, M. (2013) "An experimental comparison of periodic timetabling models". *Computers & Operations Research*, vol. 40(10): 2251–2259.

Small, K. A. (1982) "The scheduling of consumer activities: Work trips". *The American Economic Review*, vol. 72(3): 467–479.

Stern, R. (1996) "Passenger transfer system review. synthesis of transit practice". *Transportation Research Board*, vol. 19.

Törnquist, J. (2007) "Railway traffic disturbance management–an experimental analysis of disturbance complexity, management objectives and limitations in planning horizon". *Transportation Research Part A: Policy and Practice*, vol. 41(3): 249–266.

van den Heuvel, A., van den Akker, J. and van Kooten, M. (2008) "Integrating timetabling and vehicle scheduling in public bus transportation". Tech. rep., UU-CS-2008-003. Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands.

Wagner, J. L. and Falkson, L. M. (1975) "The optimal nodal location of public facilities with price-sensitive demand". *Geographical Analysis*, vol. 7(1): 69–83.

Chapter 3

On-line vehicle rescheduling in a transit line

ABSTRACT
Public transportation systems in metropolitan areas carry a high density of traffic daily, heterogeneously distributed, and exposed to the negative consequences derived from service disruptions. Breakdowns, accidents, strikes, require on-line operation adjustments to address these incidents in order to reduce their side effects, such as passenger extra-waiting times, complaints, potential operational dangers, etc. The Rescheduling Problem consists of defining a new schedule for a set of previously scheduled trips, given that one/several trips cannot be carried out. This chapter deals with the rescheduling problem in a transit line that has suffered a fleet size reduction. We present different modelling possibilities depending on the assumptions that need to be included in the modelization and we show that the problem can be solved rapidly by using a constrained max-cost-flow problem whose coefficient matrix we prove is totally unimodular. We test our results in a testbed of random instances outperforming previous results in the literature. An experimental study, based on a line segment of the Madrid Regional Railway network, shows that the proposed approach provides optimal reassignment decisions within computation times compatible with real-time use.

Keywords: Urban transportation network; Disturbance management; Real time rescheduling.

3.1 Introduction

Public transport systems often encounter disruptions that prevent them from operating as planned. Among the examples of possible disruptions there are fleet size reductions due to breakdowns, drivers' strikes or vehicle reallocations to reinforce other sections of the transit network. To address these incidents, on-line operation adjustments are required in order to reduce the side effects of emergency incidents, such as passenger waiting/traveling times, complaints, potential operational dangers, etc. The Rescheduling Problem consists of defining a new schedule for a set of previously scheduled trips, given that one/several trips cannot be carried out. While many objectives and constraints remain from the timetabling problem, new requirements and objectives arise in this context. In terms of transportation of people, the main decisions concern the minimization of the deviations from the initial timetable in operation. This is carried out cancelling some services and/or providing new reference times for some vehicles located at specific points in the network. Further decisions may concern re-allocating other available resources. In this chapter, we address the Rescheduling Problem in a transit line.

Timetable design is a central problem in transportation planning with many interfaces with other classical problems: line planning, vehicle scheduling, and vehicle rescheduling. The Transit Network Timetabling and Scheduling Problem (TNTSP) is devoted to obtaining and optimizing departure and arrival times for each line run (trip from an origin to a final destination) to and from each station over a planning horizon imposing/optimizing different constraints and objectives. The TNTSP is based on the following general input: An infrastructure of a transport system described by a node set (network stations) and an edge set (roads/tracks between adjacent stations), a trip demand matrix between pairs of nodes of the infrastructure, a set of transit lines with associated frequencies which have already been determined in order to satisfy such trip demand and, finally, a vehicle fleet with specific characteristics. The objective of the TNTSP consists of finding arrival and departure times of each vehicle at each station such that the demand satisfaction, required fleet size and vehicle capacities can be optimized/bounded.

Accidents, strikes and other sources of vehicle delays or cancellations force to modify the scheduled timetable when vehicles in some sections cannot run according to the initial planning. Here disturbances are relatively small perturbations of the transit network that can be handled by modifying the timetable, but without modifying the duties for vehicles and drivers. Disruptions are relatively large incidents, requiring both the timetable and the duties for vehicles to be modified. There are several examples of possible disruptions that demand the rescheduling of vehicles: (1) interruptions coming from severe weather conditions, accidents, and the blockage of road or tracks sections or (2) fleet size reductions coming from vehicle breakdowns, drivers' and crew strikes (van Exel and Rietveld, 2001) or vehicle reallocations made to reinforce other sections of the transit network (Burdett and Kozan, 2009). In particular, a scheduled timetable may become infeasible simply due to a heavy passenger flow (Mesa et al., 2009b).

Rescheduling is the process of updating an existing production plan in response to disturbances or disruptions (Vieira et al., 2003). Customers plan their trips based on a known timetable, and can be greatly inconvenienced if the service does not arrive or depart at the expected time. When a disturbance occurs, like a vehicle breakdown in a certain line, the system operator must make a decision about rescheduling the remainder vehicles which are normally operating along the network in order to reduce the loss of service quality perceived by the users. An important difference between the planning stage and the rescheduling stage during disruptions is that in the latter less time is available for making decisions. In principle, solutions are expected within minutes (on-line). For the resources, another important difference is that in general there is less flexibility in the rescheduling stage, since many resource duties have already started at the time of the disruption when the rescheduling is carried out, and cannot be easily diverted. In addition, the solution space is bounded by the remaining time until the end of the rescheduling horizon, which is usually the end of the day. Hence, if the disruption happens in the evening, then the solution space is much smaller than in case the disruption happens in the morning. For example, a straight forward *myopic* strategy consists in canceling those services that serve to the least number of users. This methodology would not introduce any change/delay in the remaining timetables. Nevertheless, a recent paper by (Mesa et al., 2013) have shown that if real time control strategies are applied along a transit corridor (i.e., by allowing delays at some services of the initial schedules), then the demand satisfaction after rescheduling can be increased significantly.

Example 3.1- A minimal toy example of a rescheduling problem might be considered as follows. Let s be a station in a directed transit line where the demand pattern of total arrivals to station s is represented in Figure 3.1. We assume that initially two vehicles are scheduled departing at times t_1 and t_2, therefore, a demand $0.8 * d$ is served at time t_1 and a demand $0.2 * d$ is served at time t_2. If the transport manager has to reschedule the service establishing a single vehicle departure from station s he may decide among 3 options for this isolated scenario: (1) to keep the service that departs at t_1 and cancel the one of t_2, (2) to keep the service that departs at t_2 and cancel the one of t_1 or (3) to delay the service that departs at t_1 within time interval (t_1, t_2) and cancel the service departing at t_2. The first and second options affect to the 20% and 80% of the demand respectively. The third option represents a trade-off between the first two. In the next sections we will extend this example to reschedule a complete timetable in a transit line assuming a more complex demand pattern.

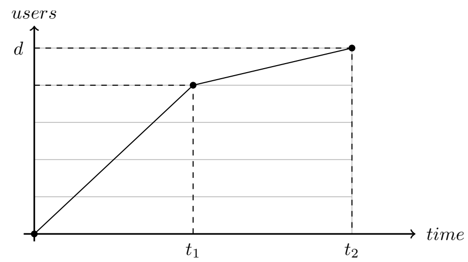


Figure 3.1: Demand pattern of total arrivals to station s.

In this chapter, we address the Rescheduling Problem in a transit line that has suffered a fleet size reduction. We describe a demand pattern to reflect the passengers' behaviour when some vehicle services are delayed or cancelled and this pattern will let us to derive a rescheduling framework coming from a timetabling formulation. We present different modelling approaches depending on the assumptions that need to be included in the modelization and we show that the problem can be rapidly solved by using a constrained max-cost-flow problem whose coefficient matrix we prove is totally unimodular. We show that our approach can be applied to real scenarios as it is the case of the commuter train systems of Madrid.

The remainder of the chapter is organized as follows. Section 3.2, reviews the most relevant contributions related to this work. Section 3.3, presents the description of the problem and all details to compute the demand pattern in the scheduling and rescheduling phase. Section 3.4, presents the rescheduling formulations that can be obtained from a general scheduling problem. Computational experiments are provided in Section 3.5 in order to show the usefulness and applicability of this methodology. Finally, some conclusions are summarized in Section 3.6.

3.2 Background

Different approaches have been developed in the literature to tackle the rescheduling problem distinguishing between (1) disturbances and disruptions, (2) the level of detail considered in the railway system, in particular in the timetable and (3) focusing the objective on the vehicles or on the customers. In the the second distinction two approaches can be distinguished, known as microscopic and macroscopic. The latter considers the transit network at a higher level, in which stations can be represented by nodes of a graph and roads/tracks by arcs, and the details of block sections and signals are not taken into account. In a microscopic approach these aspects are considered in detail. In the case of railway systems, most of the approaches in the literature deal with (1) disturbances affecting the railway system rather than disruptions, (2) the railway system at a microscopic level rather than at a macroscopic level and (3) minimizing the delays of trains or the number of canceled vehicles rather than minimizing the negative effects of disturbances and disruptions for passengers (see Cacchiani et al., 2014). This section is restricted to disruptions at a macroscopic level in a transit line.

Regarding to the timetabling problem in a transit line we refer to Mesa et al. (2014b) and references therein. Gathering the integration of timetables, vehicle scheduling and passenger choices, Mesa et al. (2014b) present a new approach for jointly planning timetables and vehicle schedules along a single transit line emphasizing the point of view of potential customers. A p-median based formulation is proposed for a given fleet size of vehicles. In addition, demand behavior is associated with the inclusion of closest assignment type constraints.

Control strategies like short turns, deadheads and/or express services can be implented for the timetabling adjustment in a transit linear corridor. Mesa et al. (2009a) develop an effective plan for allocating fleet frequencies at stops along a line based on three objectives: minimizing passenger overload, maximizing passenger mobility and minimizing passenger loss. Schedules for decongesting and recovering the line are determined by means of optimization models. The methodology proposed was applied to real data of the commuter train system of Madrid. Also Kumazawa et al. (2010) aim at minimizing the dissatisfaction experienced by the passengers due to disturbances. They propose a rescheduling algorithm that calculates a value for the amount of dissatisfaction experienced by passengers due to disturbances on the Japanese railway network. In addition to a conventional passenger flow analysis, the passenger overflow, defined as the waiting time experienced by a passenger while waiting on a platform, is considered. Nakamura et al. (2011) present an algorithm for train rescheduling during disruptions which takes as input train groups, train cancelation sections, and return patterns. These factors are predetermined by the dispatchers. Here a train group consists of a set of trains that share the same assignment of rolling stock. Train cancelation sections are sections of the railway infrastructure bounded by two stations in which all trains are canceled if a disruption occurs inside the section. In case of a disruption obstructing a section of the network, the developed algorithm determines a new timetable by canceling trains, combining return patterns, and changing the train departure order at stations in a series of steps. The efficiency of the rescheduling plan is evaluated in terms of passenger dissatisfaction caused by propagated delays. The algorithm is tested on a railway line in a metropolitan area in Japan. Sato et al. (2013) presents a timetable rescheduling algorithm based on Mixed Integer Programming (MIP) formulation when train traffic is disrupted. They minimize further inconvenience to passengers instead of consecutive delays caused by the disruption, since loss of time and satisfaction of the passengers are considered implicitly and insufficiently in the latter optimization. They presume that inconvenience of traveling by train consists of the traveling time on board, the waiting time at platforms and the number of transfers. Hence, the objective function is calculated on the positive difference between the inconvenience which each passenger suffers on his/her route in a rescheduled timetable and that in a planned timetable. The inconvenience-minimized rescheduling is often achieved at the cost of further train delays. Some trains dwell longer at a station to wait for extra passengers to come or to keep a connection, for instance.

Mesa et al. (2013) and Mesa et al. (2014a) assess the decision of rescheduling a train service, reducing the current supply along one transportation line in order to reinforce the service of another line, exploited by the same public operator, which has suffered an incidence or emergency. A methodology, based on a geometric representation of solutions which allows the use of discrete optimization techniques, is developed in order to attend to the underlying demand with efficiency criteria in this context of unexpected incidents. The proposed methodology is computationally tested and applied to real data of the commuter train system of Madrid.

This chapter differs from all previous cited research in several facets that we believe provide a significant contribution in this field. First, we give a description of the problem providing an users' demand pattern for modeling both the arrival pattern and the passengers' inconvenience function after the rescheduling. This setting extends the one in Mesa et al. (2013) and Mesa et al. (2014a) to a general framework in a transit line. Second, a modelling framework for rescheduling the line is derived from a scheduling formulation. We present different modelling approaches depending on the assumptions that need to be included in the modelization and we show that the problem is equivalent to a constrained max-cost-flow problem whose coefficient matrix we prove is totally unimodular. We test our results in a testbed of random instances outperforming previous results in the literature. An experimental study, based on a line segment of the Madrid Regional Railway network, shows that proposed approach provides optimal reassignment decisions within computation times compatible with real-time use.

3.3 Problem description

3.3.1 Infrastructure

Let l be a directed transit line running along a set of stations $s \in S$. Each station $s \in S$ also occupies the position s along the transit line l. We denote by $\langle s \rangle$ the "name" of station s that it could be a text string (e.g. $\langle 4 \rangle$ = "central station") or a number (e.g. $\langle 4 \rangle$ = "312"). Each vehicle $k \in K$ ($|K| = \kappa$) operates along l during a time horizon that will be discretized into a set of time slots

$t \in T = \{1, ..., |T|\}$ performing a single line run or expedition along the line. In addition, all vehicles will be assumed to have the same circulation speed (and unlimited capacity), so overtaking is not allowed.

3.3.2 Demand

Passengers enter in a station s and wait until a vehicle arrives. Let a_{st} be the number of passengers that access to station s at time t. A passenger that arrives to station s at time t is served by the next vehicle that departs (strictly) after time $t-1$ (denoted by vehicle $k_{st} \in K$). Since vehicles are assumed to have unlimited capacity, once a vehicle leaves a station all passengers waiting at the station leave with it. We assume that passengers that entered to a station at time t' suffer an inconvenience $\varphi_{st't} \in [0,1]$ if they have to wait until time $t > t'$ for leaving/departing. Without loss of generality we can set the inconvenience equal to zero if t is not greater than an amount $\tau^1_{s,t'}$. This means that passengers may wait a certain amount of time without suffering any inconvenience. On the other hand, we can assume that the inconvenience is the maximum ($\varphi_{st't} = 1$) after a time $\tau^2_{s,t'} < t$. Inside interval $(\tau^1_{s,t'}, \tau^2_{s,t'})$ the inconvenience is assumed to take a value $\alpha_{st't} \in [0,1)$. In this way, φ takes the following expression:

$$\varphi_{st't} = \begin{cases} 0, & t' < t \leq \tau^1_{st'}; \\ \alpha_{st't} \in [0,1), & \tau^1_{st'} \leq t < \tau^2_{st'}; \\ 1, & \tau^2_{st'} \leq t; \end{cases} \tag{3.1}$$

Example 3.2- Figure 3.2-left shows an example of constant arrivals pattern. The total number of users waiting at station s during time interval $[t', t]$ (that is $\sum_{t\in[t',t]} a_{st}$) is also depicted. This pattern may correspond to a situation where users did not know vehicle departure times. Figure 3.2-center shows another example of arrivals pattern (concentrated around time slot $t'+4$). This pattern may correspond to a situation where users know that a vehicle departure would take place at time $t'+7$. Figure 3.2-right shows an example of inconvenience cost function for those users that arrived at time slot t' and started suffering an inconvenience after time $\tau^1_{s,t'}$. Note that at time t, with $t' < t \leq \tau^1_{s,t'}$, the inconvenience remains constant with value equal to 0 and at time t, with $(t \geq \tau^2_{s,t'} = \tau^1_{s,t'} + 6$, the inconvenience remains constant with value equal to 1.

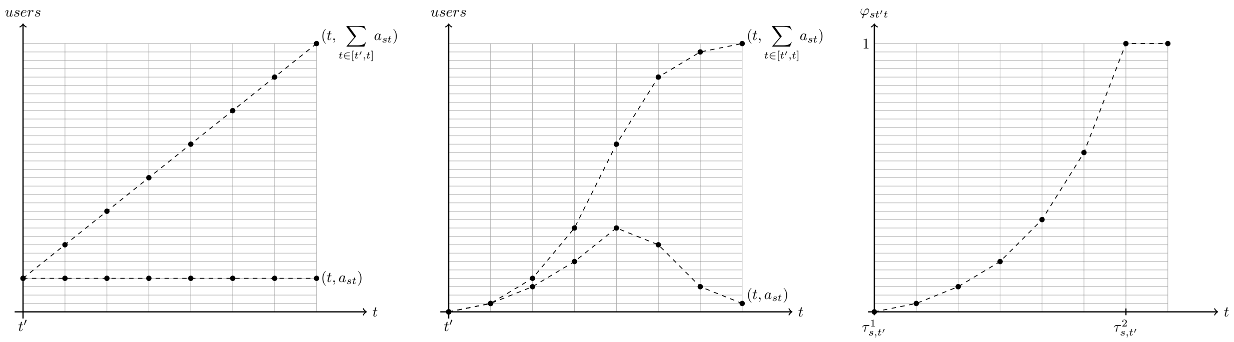


Figure 3.2: Demand patterns (left, center) and inconvenience function (right).

3.3.3 Timetables

The concept of timetable can be formalized as follows. Given the set of vehicles $k \in K$ defined in line l, a timetable Θ along partition T is defined as the set of arrival/departure times at each station for each vehicle: $\Theta = \{(\theta^+_{sk}, \theta^-_{sk}), s \in S, k \in K\}$ (for modeling convenience we can assume that $\theta^+_{1k} = \theta^-_{1k} - 1$, $\theta^-_{|S|k} = \theta^+_{|S|k} + 1$). Denoting by λ_{sk} the waiting time of vehicle k at station $s \in S$ and by μ_{sk} the travel time from station s to the next station (that is, the travel time between stations s and $s+1$) we can assume that:

1. $\lambda_{sk} = \theta^-_{sk} - \theta^+_{sk}$
2. $\mu_{sk} = \theta^+_{s+1,k} - \theta^-_{sk}, \quad s \in S : s < |S|, k \in K$

A timetable Θ defined by variables $\theta^+, \theta^-, \lambda$ and μ is called *arrival-departure timetable*. In addition we denote by λ^* (λ_*) the maximum (minimum) waiting time that a vehicle can stay in a station that is not a terminal of the transit line.

Potentially, all timetables can be generated over sets S, K. However, the number of feasible timetables can be highly reduced by means of the following result:

Property 3.1- A timetable Θ can be redefined as $\Theta \equiv x = \{x_{st}, t \in T, s \in S : s < |S|\}$ where $x_{st} \in \{0,1\}$ is equal to 1 if and only if a vehicle departs from station s at time t and:

- The departure time of vehicle k from station s: $\theta^-_{sk} = \{t : x_{st} = 1, t \in T\}|_k$ (where $\cdot|_k$ denotes the k-th element of a set that is sorted in non decreasing order).
- The arrival time of vehicle k to station s: $\theta^+_{sk} = \theta^-_{s-1k} + \mu_{s-1k}$.
- The waiting time of vehicle k at station $\lambda_{sk} = \theta^-_{sk} - \theta^+_{sk}$.

A timetable Θ defined by variables x is called *departure timetable*.

Remark 3.1- Since no passengers wait at station $|S|$, without loss of generality we can exclude in the following such station from the set S.

Property 3.2- On each station we consider a discretized version of the time horizon in time slots (typically in minutes) $t \in T_s \subset T$. The set of time slots T_s avoids departures at times that are too late to reach the end of the line or too early if such station has to be reached from the beginning of the line. In order to compute T_s for each $s \in S$ we assume that a minimum waiting time $\lambda_* = 1$ is required at each station.

Example 3.3- Table 3.1 shows an arrival-departure timetable for two vehicles in a directed transit line running along stations 12, 7, 9, 13 that occupy positions 1, 2, 3 and 4 in the line, respectively. We also show the departure timetable (in terms of variables x). Given $|T| = 23$ and applying Property 3.2 we can reduce the feasible time slots for each station as it is indicated.

		$k=1$			$k=2$				
$\langle s \rangle$	s	$(\theta^+_{sk}, \theta^-_{sk})$	λ_{sk}	μ_{sk}	$(\theta^+_{sk}, \theta^-_{sk})$	λ_{sk}	μ_{sk}	$x = (x_{st})$	T_s
$\langle 12 \rangle$	1	(1,2)	1	2	(1,8)	7	2	$(x_{1t}) : x_{1t} = 1, t \in \{2, 8\}; x_{1t} = 0, o.c.$	$\{2, ..., 11\}$
$\langle 7 \rangle$	2	(4,7)	3	4	(10,12)	2	4	$(x_{2t}) : x_{2t} = 1, t \in \{7, 12\}; x_{2t} = 0, o.c.$	$\{5, ..., 14\}$
$\langle 9 \rangle$	3	(11,13)	2	3	(16,19)	3	3	$(x_{3t}) : x_{3t} = 1, t \in \{13, 19\}; x_{3t} = 0, o.c.$	$\{10, ..., 19\}$
$\langle 13 \rangle$	4	(16,23)	7	0	(22,23)	1	0		

Table 3.1: Two representations of a timetable in a directed transit line.

Next, we show how travel times between stations can be assumed to be all equal to a constant (if all vehicles travel at the same speed). We will make this assumption in the following.

Property 3.3- Travel times between stations can be all considered equal to a constant (that is, $\mu_{sk} = 1,\ s \in S, k \in K$).

`Proof.`

Since it is assumed that all vehicles travel at the same speed, we just set $\theta^-_{sk} := \theta^-_{sk} - \sum_{s' \in S : s' < s} \mu_{s'k},\ s \in S : 1 < s < |S|$. □

Example 3.4- Table 3.2 shows the arrival-departure timetable of example 1 when travel times are equal to 1 (in the x variables). Note that now we can reduce the feasible time slots for each station (T_s) as it is indicated.

		$k=1$			$k=2$				
s	s	$(\theta^+_{sk}, \theta^-_{sk})$	λ_{sk}	μ_{sk}	$(\theta^+_{sk}, \theta^-_{sk})$	λ_{sk}	μ_{sk}	x	T_s
$\langle 12 \rangle$	1	(1,2)	1	1	(1,8)	7	1	$(x_{st}) : x_{st} = 1, t \in \{2, 8\}; x_{st} = 0, o.c.$	$\{2, ..., 10\}$
$\langle 7 \rangle$	2	(3,6)	3	1	(9,11)	2	1	$(x_{st}) : x_{st} = 1, t \in \{6, 11\}; x_{st} = 0, o.c.$	$\{4, ..., 13\}$
$\langle 9 \rangle$	3	(7,9)	2	1	(12,15)	3	1	$(x_{st}) : x_{st} = 1, t \in \{9, 15\}; x_{st} = 0, o.c.$	$\{6, ..., 15\}$
$\langle 13 \rangle$	4	(10,17)	7	0	(16,17)	1	0		

Table 3.2: Two representations of a timetable in a directed transit line.

Remark 3.2- The previous settings remains valid for representing timetables in the case of a bi-directional line (a transit corridor with two lines, on each direction, running along all stations). For doing so, we replicate each station once each time it is visited in the timetable so if vehicles run back and forth along the line n times, we can redefine set S as:

$$S := \{1, ..., |S|, |S| + 1, ..., 2|S|, ..., (2n-2)|S|, ..., (2n-1)|S|, (2n-1)|S|, ..., 2n|S|\}.$$

Example 3.5- Let L be a transit corridor with 5 stations $\{a, b, c, d, e\}$ ($|S| = 4$) where vehicles run back and forth 2 times. Then, redefining set S, we can assume that a single directed line runs along L passing through stations:

$\langle s \rangle$	a	b	c	d	e	d	c	b	a	b	c	d	e	d	c	b
$s \in S$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Table 3.3: Relation between a bidirectional transit line traversed four times and a directed transit line of 16 stations.

3.3.4 Inconvenience cost function under disruptions

In this section we assume that a number of κ vehicles was initially scheduled within timetable Θ. If a subset of vehicles becomes unavailable, a new set of $\bar{\kappa}$ departure times at each station has to be determined where $\bar{\kappa} < \kappa$. Passengers ignore new departure times until they arrive to a station at a time t. We denote by $k_{st'}$ the first vehicle, of the original schedule, with a departure after t' from station s and by $\theta^-_{s,k_{st'}}$ such departure time. Three possible decisions must be taken for each departure time $\theta^-_{s,k_{st'}}$ (or service) initially scheduled:

1. keep the service in the initial timetable,
2. delay the service within time interval $(\theta^-_{s,k_{st'}}, \theta^-_{s,k_{st'}+1})$ or
3. cancel the service.

In this way, the inconvenience suffered by passengers arriving at time t' will be 0 if they depart from s as in normal operation, that is, no later than $\theta^-_{s,k_{st'}}$. Otherwise, if the departure from s is at a time t within time interval $(\theta^-_{s,k_{st'}}, \theta^-_{s,k_{st'}+1})$, passengers arriving at time t' will suffer an inconvenience given by a value $\varphi_{st't} = \alpha_{st't} \in [0, 1)$. In addition, to penalize cancelled services, we can assume that the inconvenience for passengers arriving at time t' is full if service $\theta^-_{s,k_{st'}}$ is cancelled. Therefore, the inconvenience function under disruptions is given by $\varphi_{st't}$ when $\tau^1_{st'} = \theta^-_{s,k_{st'}}$ and $\tau^2_{st'} = \theta^-_{s,k_{st'}+1}$:

$$\tilde{\varphi}_{st't} = \begin{cases} 0, & t' < t \leq \theta^-_{s,k_{st'}}; \\ \alpha_{st't} \in [0,1), & \theta^-_{s,k_{st'}} \leq t < \theta^-_{s,k_{st'}+1}; \\ 1, & \theta^-_{s,k_{st'}+1} \leq t; \end{cases} \tag{3.2}$$

3.4 Problem formulation

In this section we present a catalogue of valid formulations for the problem described in the previous sections. We begin with a scheduling formulation F^{xy} that is valid for any inconvenience function $\varphi_{st't}$.

Next, we will use the inconvenience cost function under disruption $\tilde{\varphi}_{st't}$ in order to penalize delayed vehicles and, more strongly, cancelled services. We show that $\tilde{\varphi}_{st't}$ allows us to reformulate F^{xy} into an equivalent formulation easier to solve. In this way, 2 different formulations are presented for the rescheduling problem depending on including explicitly or not arrival departures times to stations.

3.4.1 Scheduling formulation (F^{xy})

In the following, we consider scheduling, timetabling and vehicle scheduling as synonyms since each vehicle will perform a single line-run along the transit line. Therefore we assume that κ vehicles are scheduled along the line and the demand arriving to station s at time t' is assigned to the first vehicle departing from that station after time t'. We recall that x_{st} is defined as a binary variable equal to 1 if a vehicle departs from station s at time t. In addition we require a binary variable that assigns passengers to a time slot where there must exist a vehicle departure. Then, let $y_{st't}$ be a binary variable equal to 1 if passengers arriving at time t' are allocated to a vehicle departing at time t.

$$F^{xy}: \quad \min \sum_{s\in S}\sum_{t'\in T_s}\Big(\sum_{t\in T_s:t'<t} a_{st'}\varphi_{st't}y_{st't} + a_{st'}\big(1-\sum_{t\in T_s:t'<t} y_{st't}\big)\Big) \tag{3.3a}$$

$$s.t. \quad \sum_{t\in T_s} x_{st} = \kappa \qquad s\in S \tag{3.3b}$$

$$\sum_{t'\in T_s:t'\le t} x_{st'} \le \sum_{t'\in T_{s+1}:t'\le t+\mu_s+\lambda^*} x_{s+1t'} \qquad s\in S, t\in T_s: s<|S| \tag{3.3c}$$

$$x_{st}\in\{0,1\} \qquad s\in S, t\in T_s \tag{3.3d}$$

$$y_{st't}\le x_{st} \qquad s\in S, t',t\in T_s: t'\le t \tag{3.3e}$$

$$\sum_{t\in T_s:t>t'} y_{st't}\le 1 \qquad s\in S, t'\in T_s \tag{3.3f}$$

$$y_{st't}\in\{0,1\} \qquad s\in S, t',t\in T_s: t'\le t \tag{3.3g}$$

The objective function (3.3a) minimizes the total users' inconvenience. It indicates that the inconvenience of passengers that arrived to station s at time t' is $\varphi_{st't}$ if they are allocated to a vehicle departing at time t. Otherwise, if demand $a_{st'}$ is not allocated to any time slot, the inconvenience for those passenger will be full. Constraints (3.3b) impose that at each station all vehicles have to depart (that is, there are κ departures and, in total, κ line runs along the line). Constraints (3.3c) ensure flow conservation of vehicles at stations, imposing that the number of vehicles departing from station s before time t is lower than those departing from the next station before time $t+\mu_s+\lambda^*$. Constraints (3.3e) ensure that no passenger allocations are made to timetables that do not exist. Constraints (3.3f) impose that each demand $a_{st'}$ is allocated to no more than one line run.

Formulation F^{xy} comes from the scheduling formulation developed in Mesa et al. (2014b) where the timetabling problem is seen as a p-median based formulation. Here, F^{xy}, extends the flexibility of the timetable since vehicles can remain stopped at intermediate stations allowing a better adjustment of the demand. In addition, since objective (3.3a) shows how much is the global inconvenience of the scheduling, we can equivalently represent also the global convenience of the scheduling just subtracting this amount from the total number of passengers in the system. Therefore, denoting by $A = \sum_{s\in S}\sum_{t'\in T} a_{st'}$, the global convenience of the scheduling can be defined as:

$$A - \min \sum_{s\in S}\sum_{t'\in T_s}\Big(\sum_{t\in T_s:t'<t} a_{st'}\varphi_{st't}y_{st't} + a_{st'}(1 - \sum_{t\in T_s:t'<t} y_{st't})\Big) \tag{3.3a'}$$
$$s.t. (3.3b) - (3.3g)$$

Property 3.4- Objective (3.3a') is equivalent to

$$\max \sum_{s\in S}\sum_{t'\in T_s}\sum_{t\in T_s:t'<t} a_{st'}(1-\varphi_{st't})y_{st't} \tag{3.3a"}$$

`Proof.`
To prove comes straightforward from transforming (3.3a') into (3.3a")

$$A - \min \sum_{s\in S}\sum_{t'\in T_s}\Big(\sum_{t\in T_s:t'<t} a_{st'}\varphi_{st't}y_{st't} + a_{st'}(1 - \sum_{t\in T_s:t'<t} y_{st't})\Big)) =$$
$$= A - \min \sum_{s\in S}\sum_{t'\in T_s}\Big(\sum_{t\in T_s:t'<t} a_{st'}\varphi_{st't}y_{st't} + a_{st'} + \sum_{t\in T_s:t'<t} -a_{st'}y_{st't})\Big)) =$$
$$= A - \min \sum_{s\in S}\sum_{t'\in T_s}(a_{st'} + \sum_{t\in T_s:t'<t} a_{st'}(\varphi_{st't}-1)y_{st't}) =$$
$$= A - \min \sum_{s\in S}\sum_{t'\in T_s}(a_{st'} - \sum_{t\in T_s:t'<t} a_{st'}(1-\varphi_{st't})y_{st't}) =$$
$$= A - \min(A - \sum_{s\in S}\sum_{t'\in T_s}\sum_{t\in T_s:t'<t} a_{st'}(1-\varphi_{st't})y_{st't}) =$$
$$= A + \max(-A + \sum_{s\in S}\sum_{t'\in T_s}\sum_{t\in T_s:t'<t} a_{st'}(1-\varphi_{st't})y_{st't}) =$$
$$\max \sum_{s\in S}\sum_{t'\in T_s}\sum_{t\in T_s:t'<t} a_{st'}(1-\varphi_{st't})y_{st't}$$

□

3.4.2 Rescheduling formulation (F^{st})

In this section, we aim to derive a rescheduling formulation valid to reschedule a timetable (for example, a timetable generated with formulation F^{xy} or with the one in Mesa et al., 2014b). In order to compare results with previous ones in the literature, we (equivalently) change the objective function from a minimization of users' inconvenience to a maximization of users' convenience, where we understand the convenience cost function as $1 - \tilde{\varphi}$.

$$F^{st}: \quad \max \sum_{s \in S} \sum_{t' \in T_s} \sum_{t \in T_s : t' < t < \theta^-_{s,k_{s,t'}+1}} a_{st'}(1 - \tilde{\varphi}_{st't}) x_{st} \tag{3.5a}$$

$$s.t. \quad \sum_{t \in T_s} x_{st} = \bar{\kappa} \qquad s \in S \tag{3.5b}$$

$$x_{st} \leq \sum_{t' \in T_s : t + \mu_s + \lambda_* \leq t' \leq t + \mu_s + \lambda^*} x_{s+1t'} \qquad s \in S : s < |S|, t \in T_s \tag{3.5c}$$

$$\sum_{t \in T_s : \theta^-_{s,k-1} < t < \theta^-_{s,k+1}} x_{st} \leq 1 \qquad s \in S, k \in K \tag{3.5d}$$

$$x_{st} \in \{0, 1\} \qquad s \in S, t \in T_s \tag{3.5e}$$

The objective function (3.5a) maximizes the total users' convenience. It indicates that the convenience of passengers that arrived to station s at time t' is $1 - \varphi$ if a vehicle departs at time t. Note that out from interval $(t', \theta^-_{s,k_{s,t'}+1})$ the convenience for demand $a_{st'}$ is not defined but it results in being computed as 0. Note also that (3.5a) is well defined since constraints (3.5d) ensure that no more than one vehicle is rescheduled inside interval $(\theta^-_{s,k-1}, \theta^-_{s,k+1})$ avoiding demand $a_{st'}$ being served by more than one vehicle. Constraints (3.5c) impose that if a vehicle departs from station s at time t, then another vehicle must depart from the next station inside time interval $[t + \mu_s + \lambda_*, t + \mu_s + \lambda^*]$ (we recall $\theta^-_{s,0} = 0$, $\theta^-_{s,\kappa+1} = |T_s| + 1$.

The reader, may note that F^{st} generates a rescheduled timetable that is the same as the one of F^{xy} when $\varphi = \tilde{\varphi}$. However, formulation F^{st} presents other modelling advantages and solution possibilities that we describe in the following section. In terms of location theory, F^{xy} presents a location-allocation problem (in terms of the x and y variables respectively), whereas F^{st} can be seen as a maximum covering problem (Church and ReVelle, 1974) where each located timetable x_{st} covers a certain demand, but no demand is covered more than once.

3.4.3 Rescheduling formulation (F^{sut})

In this section we specify in the x variables the time at which a vehicle arrives and departs from a station. We define as T_s^2 the set of time slot pairs that are feasible for arriving and departing at a station $s \in S$. In particular, we define T_s^2 as

$$T_s^2 = \begin{cases} \{(t-1,t) : u = 1, t \in T_1\}, & \text{iif } s = 1; \\ \{(u,t) : u - \mu_{s-1} \in T_{s-1}, t \in T_s, u + \lambda_* < t < u + \lambda^*\} & \text{iif } s > 1; \end{cases} \tag{3.6}$$

Then, we denote by x_{sut} the binary variable equal to 1 if coordinates (u, t) of the time map corresponding to station s are occupied by a vehicle; 0 otherwise. The reader may check Mesa et al. (2013) for further details on timetable representations as arrival/departure diagrams.

$$F^{sut}: \quad \max \sum_{s \in S} \sum_{t' \in T_s} \sum_{(u,t) \in T_s^2 : t' < t < \theta^-_{s,k_{s,t'}+1}} a_{st'}(1 - \tilde{\varphi}_{st't}) x_{sut} \tag{3.7a}$$

$$s.t. \quad \sum_{(u,t) \in T_s^2} x_{sut} = \kappa \qquad s \in S \tag{3.7b}$$

$$\sum_{(u,t') \in T_s^2 : t' \leq t} x_{sut'} \leq \sum_{(u,t') \in T_{s+1}^2 : t' \leq t + \mu_s} x_{s+1ut'} \qquad s \in S, t \in T_s : s < |S| \tag{3.7c}$$

$$\sum_{(u,t) \in T_s^2 : \theta^-_{s,k-1} < t < \theta^-_{s,k+1}} x_{sut} \leq 1 \qquad s \in S, k \in K \tag{3.7d}$$

$$x_{sut} \in \{0, 1\} \qquad s \in S, (u,t) \in T_s^2 \tag{3.7e}$$

The objective function (3.7a) maximizes the total users' convenience. Constraints (3.7b) impose that on each station all vehicles are scheduled. Constraints (3.7c) impose that the number of vehicles departing from station s before time $t \in T_s$, has to be lower or equal to the number of vehicles departing from station $s+1$ at time $t + \mu_s$. Constraints (3.7d) impose that there is no more than one vehicle departing from station s inside time interval $(\theta^-_{s,k-1}, \theta^-_{s,k+1})$.

Note that each solution of formulation F^{sut} is related with a solution of F^{st} by means of the relationships $x_{st} = \sum_{u \in T_s : u < t} x_{sut}$ and

$$x_{sut} = \begin{cases} 1, & \text{if } s = 1, x_{st} = 1, u = t - 1; \\ 1, & \text{if } s > 1, x_{st} = 1, x_{s-1,u-\mu_{s-1}} = 1; \\ 0, & \textit{otherwise}; \end{cases} \tag{3.8}$$

Property 3.5- F^{sut} can be solved by by using linear programming.

`Proof.`

We base our proof in two steps: (1) We prove that F^{sut} is equivalent to another problem $\vec{F}^{sut}$ and (2) denoting by $\vec{A}^{sut}$ to the matrix of coefficients coming from problem $\vec{F}^{sut}$, we prove that $\vec{A}^{sut}$ is totally unimodular (TU).

1. Let $\vec{x}_{sut}$ be a binary variable equal to 1 if a vehicle arrives at time u to station s and departs at

time t. Note that $t \in [u+\lambda_*, u+\lambda^*]$. We give the proof for the general case when at each s, $\mu_s = 0$, $\lambda_* = 0$ and $\lambda^* = |T_s| - t$. Let $\vec{F}^{sut}$ be the following formulation:

$$\vec{F}^{sut}: \quad \max \sum_{s\in S}\sum_{t'\in T_s}\sum_{t\in T_s: t'<t<\theta^-_{s,k_{s,t'}+1}} a_{st'}(1-\tilde{\varphi}_{st't})x_{st} \tag{3.9a}$$

$$s.t. \quad \sum_{t\in T_s} x_{1t} = \bar{\kappa} \tag{3.9b}$$

$$x_{s-1,u} - \sum_{t\in T_s: t>u} \vec{x}_{sut} = 0 \qquad s\in S: 1<s<|S|, u\in T_s \tag{3.9c}$$

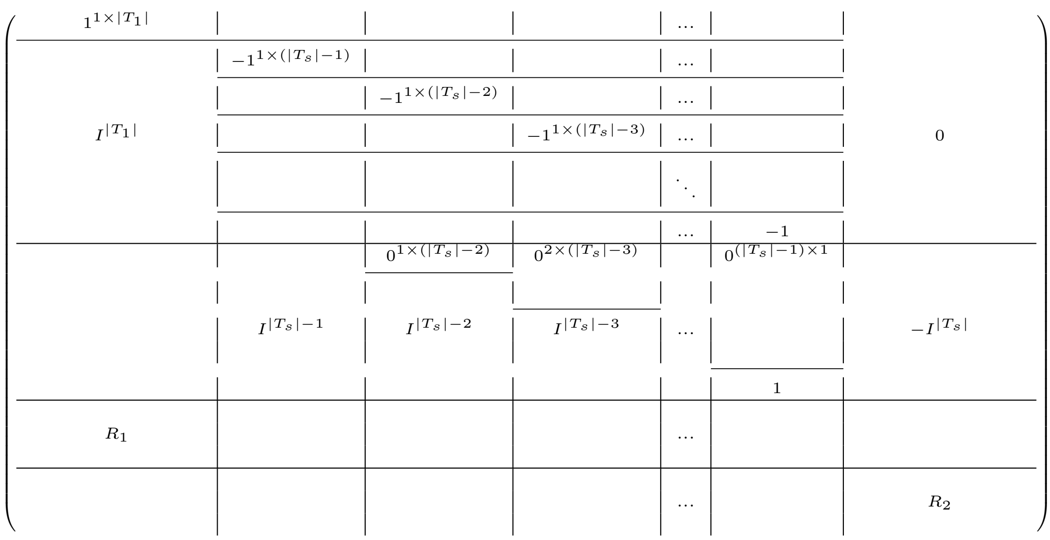
$$-x_{s,t} + \sum_{u\in T_s: u<t} \vec{x}_{sut} = 0 \qquad s\in S: s<|S|, t\in T_s \tag{3.9d}$$

$$\sum_{t\in T_s: \theta^-_{s,k-1}<t<\theta^-_{s,k+1}} x_{st} \le 1 \qquad s\in S, k\in K \tag{3.9e}$$

$$x_{st}\in\{0,1\} \qquad s\in S, t\in T_s \tag{3.9f}$$

Constraints (3.9c)–(3.9d) are equivalent to (3.5c) so the solution space in terms of variables x_{st} is the same for F^{sut}, F^{st} and F^{sut}. Problem (3.9a)–(3.9f) can be seen a constrained maximum cost flow problem in a directed network where $\bar{\kappa}$ units of flow (vehicles) are sent from station 1 to station $|S|$. Arcs can be considered as trips between adjacent stations at a time instant and the cost of each edge is the captured demand when a vehicle departs at a certain time.

2. Let $\vec{A}^{sut}$ be the matrix of coefficients coming from problem $\vec{F}^{sut}$, we prove that $\vec{A}^{sut}$ is totally unimodular. Note first that $(\vec{A}^{sut})_{s\in\{1,2\},u,t\in T_s:u\le t}$ has the following form:



where empty boxes represent zeros, $1^{m\times n}$ stands for a matrix $m\times n$ of ones (analogously for 0 and -1), I^n is the identity matrix in dimension n and R_s is a stair-matrix of consecutive ones in the rows and columns with no more than two ones per column.

To prove that $\vec{A}^{sut}$ is TU we give the argument for $(\vec{A}^{sut})_{s\in\{1,2\},u,t\in T_s:u\leq t}$ and remains the same for $\vec{A}^{sut}$. Its known that a matrix A is TU $\Leftrightarrow$ for every $J \subseteq N = (1,...,n)$ there exists a partition J_1, J_2 of J such that

$$|\sum_{j\in J_1} a_{ij} - \sum_{j\in J_2} a_{ij}| \leq 1 \forall i = 1,...,m. \tag{3.10}$$

We choose an arbitrary (sorted) subset of columns of $\vec{A}^{sut}$, $J = (j_1,...,j_l)$ and for each $j \in J$ we construct J_1 and J_2 as follows. We start with $J_1' = j_1$ and $J_2' = \emptyset$. Iteratively, we try if (3.10) is fulfilled with $J_1 = J_1' \cup j_i$ and $J_2 = J_2'$. If so, we redefine $J_1' := J_1' \cup j_i$ and otherwise, we redefine $J_2' := J_2' \cup j_i$. Doing so for each $j \in J$ we conclude defining $J_1 = J_1'$ and $J_2 = J_2'$.

□

3.4.4 Extensions

The presented models assume that all vehicles are identical. This assumption is reasonable but could be too restrictive in practical situations. Therefore, further side constraints can be added to F^{stk} and F^{sutk} for example in terms of capacities and travel times. It is out of the scope of this chapter to discuss all these possible considerations but it might be of interest to include the extensions of formulations F^{st} and F^{sut} to the case when we make distinction among the different vehicles.

Let x_{stk} be a binary variable equal to 1 if vehicle $k \in \bar{K}$ (we recall $\bar{K}$ is the set of vehicles that are about to be rescheduled) departs from station s at time t (0 otherwise) and analogously x_{stuk} be a binary variable equal to 1 if vehicle k departs arrives to station s at time u and departs from s at time t. We denote these formulations F^{stk} and F^{sutk} and the can be derived straight forward from F^{st} and F^{sut} as follows:

$$F^{stk}: \quad \max \sum_{k\in\bar{K}} \sum_{s\in S} \sum_{t'\in T_s} \sum_{t\in T_s: t'<t<\theta^-_{s,k_{s,t'}+1}} a_{st'}(1-\tilde{\varphi}_{st't})x_{stk} \tag{3.11a}$$

$$s.t. \quad \sum_{t\in T_s} x_{stk} = 1 \qquad s\in S, k\in\bar{K} \tag{3.11b}$$

$$\sum_{t'\leq t} x_{st'k} \leq \sum_{t'\leq t+\mu_s} x_{s+1t'k} \qquad s\in S, t\in T_s, k\in K \tag{3.11c}$$

$$\sum_{k'\in K} \sum_{t\in T_s:\theta^-_{s,k-1}<t<\theta^-_{s,k+1}} x_{stk'} \leq 1 \qquad s\in S, k\in\bar{K} \tag{3.11d}$$

$$x_{stk} \in \{0,1\} \qquad s\in S, t\in T_s, k\in\bar{K} \tag{3.11e}$$

$$F^{sutk}: \quad \max \sum_{k\in\bar{K}} \sum_{s\in S} \sum_{t'\in T_s} \sum_{(u,t)\in T_s^2: t'<t<\theta^-_{s,k_{s,t'}+1}} a_{st'}(1-\tilde{\varphi}_{st't}) x_{sutk} \tag{3.12a}$$

$$s.t. \quad \sum_{(u,t)\in T_s^2} x_{sutk} = 1 \qquad s\in S, k\in\bar{K} \tag{3.12b}$$

$$\sum_{(u,t')\in T_s^2: t'\le t} x_{sut'k} \le \sum_{(u,t')\in T_s^2: t'\le t+\mu_s} x_{s+1ut'k} \qquad s\in S, t\in T_s, k\in\bar{K} \tag{3.12c}$$

$$\sum_{k'\in\bar{K}} \sum_{(u,t)\in T_s^2: \theta^-_{s,k-1}<t<\theta^-_{s,k+1}} x_{sutk} \le 1 \qquad s\in S, k\in K \tag{3.12d}$$

$$x_{sutk}\in\{0,1\} \qquad s\in S, (u,t)\in T_s^2, k\in\bar{K} \tag{3.12e}$$

The reader may note that the interpretation of constraints (3.11a)–(3.11e) and (3.12a)–(3.12e) is the same as the associated ones in formulations F^{st} and F^{sut}.

If no additional parameter/constraint dependant on index k is added to the model, F^{stk} contains a large set of symmetric optimal solutions ($\kappa!$) since vehicles in an optimal solution can be relabelled without changing the objective value. In this way a solution of F^{st} is related with $\kappa!$ solutions of F^{stk}. Therefore, F^{st} and F^{stk} contain the same set of non-symmetric solutions.

Formulation (3.12a)–(3.12e) is equivalent to the presented in Mesa et al. (2013) but the alternative formulation of the flow conservation constraints (3.12c) provide a better performance and a significant improvement in running times as we show in the next section.

3.5 Computational experiments

3.5.1 Testbed of random instances

In this section, the computational performance of the different formulations is assessed. We have generated similar instances to those in Mesa et al. (2013) in order to establish later a comparison with those previous results. Along a one-way transit line with a number of stations $|S| = 10$ we have generated a random instances for $|I| = 1000$ transportation requests (origin-destination trips) in the time intervals $|T| = \{60, 120, 180, 240\}$ with desired arrival times following a uniform distribution. This time-dependant origin-destination matrix has been introduced as an input to compute an optimal timetable as described in Mesa et al. (2014b). This timetable gave us the number of passengers that board at each departure time.

Each of our tables reports the following items. Each row corresponds to a group of 5 instances with the same characteristics ($|T|, \kappa, \bar{\kappa}$) indicated in the first 3 columns (we recall $|T|$ the number of time slots in the time horizon, κ the fleet size before the rescheduling and $\bar{\kappa}$ the fleet size to be rescheduled). Column $t/gap(\#)$ reports firstly the average running time in seconds of the 5 instances of the row. If any of the 5 instances was solved to optimality, this column reports the average relative gap (indicated with a percentage %) computed with the best solution found by the solver and the LP bound. In addition, if at least one instance reaches the CPU time limit, we indicate in brackets the number of instances that could be solved to optimality within the time limit and, in such a case, we compute the average running time by using the time limit for those instances that could not be solved to optimality. Column t^*/gap^* reports the biggest CPU time over the 5 instances of the group. Whenever the time limit is reached for any instance, the maximum relative gap (indicated with a percentage %) is reported instead. Column *gapLR* reports the relative gap computed with the best solution found by the solver and the optimal value of the linear relaxation at the root node. Column *nodes* indicates the average number of nodes explored in the branch and bound tree. Finally, column *obj* reports the average objective value of the 5 instances of the row. All tables report analogous items for the different formulations described along the chapter. In order to facilitate the comparison among all tables, we have marked in bold red the best result among all in the same group. In case of ties the best results have been marked in bold blue.

$\lvert T\rvert$	κ	$\bar{\kappa}$	$t/gap(\#)$	t^*/gap^*	$gapLR$	nod	obj	$t/gap(\#)$	t^*/gap^*	$gapLR$	nod	obj
			F^{st}					F^{sut}				
60	4	1	0.2	0.7	10.77	**1**	**415.6**	**0.1**	**0.1**	**0**	**1**	**415.6**
60	4	2	**0**	**0.1**	1.21	**1**	**761.2**	0.1	**0.1**	**0**	**1**	**761.2**
60	4	3	0.2	0.5	1.56	**1**	**898**	**0.1**	**0.1**	**0**	**1**	**898**
60	4	4	**0**	**0**	**0**	**1**	**1000**	0.1	0.1	**0**	**1**	**1000**
120	9	1	1.3	3	11.34	**1**	**199.6**	**0.3**	**0.4**	**0**	**1**	**199.6**
120	9	2	7.1	9.2	10.69	1081	**378.8**	**0.3**	**0.3**	**0**	**1**	**378.8**
120	9	3	10.9	18.9	8.6	1167	**541.2**	**0.3**	**0.3**	**0**	**1**	**541.2**
120	9	4	5.8	12.3	3.99	101	**690.8**	**0.3**	**0.3**	**0**	**1**	**690.8**
120	9	5	2.2	8.2	1.76	7	**794.2**	**0.3**	**0.3**	**0**	**1**	**794.2**
120	9	6	0.9	3	1.43	3	**856.6**	**0.3**	**0.3**	**0**	**1**	**856.6**
120	9	7	0.4	0.5	0.99	**1**	**908.4**	**0.3**	**0.3**	**0**	**1**	**908.4**
120	9	8	0.4	1	0.47	**1**	**956.4**	**0.3**	**0.3**	**0**	**1**	**956.4**
120	9	9	**0.1**	**0.1**	**0**	**1**	**1000**	0.3	0.3	**0**	**1**	**1000**
180	14	1	6	12.2	15.02	6	**128.8**	**0.5**	**0.5**	**0**	**1**	**128.8**
180	14	2	19.2	37.6	13.39	4076	**251.8**	**0.4**	**0.5**	**0**	**1**	**251.8**
180	14	3	151.5	489.7	11.7	33718	**365.6**	**0.4**	**0.5**	**0**	**1**	**365.6**
180	14	4	633	1384.2	10.4	220693	**471.2**	**0.5**	**0.5**	**0**	**1**	**471.2**
180	14	5	548.4	1692.9	8.95	74771	**569.8**	**0.5**	**0.5**	**0**	**1**	**569.8**
180	14	6	84.1	245.5	6	9289	**663.6**	**0.5**	**0.5**	**0**	**1**	**663.6**
180	14	7	15.2	33.5	3.21	288	**742.6**	**0.5**	**0.5**	**0**	**1**	**742.6**
180	14	8	4.1	8.4	1.94	6	**798.6**	**0.5**	**0.5**	**0**	**1**	**798.6**
180	14	9	2.2	6.6	1.42	2	**842.6**	**0.5**	**0.5**	**0**	**1**	**842.6**
180	14	10	1.4	2.4	1.26	**1**	**879.4**	**0.5**	**0.5**	**0**	**1**	**879.4**
180	14	11	1.5	2.9	1.15	**1**	**911.8**	**0.5**	**0.5**	**0**	**1**	**911.8**
180	14	12	1.3	2.7	0.78	**1**	**943.4**	**0.5**	**0.5**	**0**	**1**	**943.4**
180	14	13	0.6	0.9	0.41	**1**	**972.6**	**0.5**	**0.5**	**0**	**1**	**972.6**
180	14	14	**0.2**	**0.2**	**0**	**1**	**1000**	0.4	0.4	**0**	**1**	**1000**
240	18	1	10.3	13.6	16.31	18	**100**	**0.7**	**0.8**	**0**	**1**	**100**
240	18	2	63.4	143.6	16.06	10678	**196**	**0.6**	**0.6**	**0**	**1**	**196**
240	18	3	579.2 (4)	4.95%	15.52	159843	**286.2**	**0.6**	**0.6**	**0**	**1**	**286.2**
240	18	4	1434.3 (2)	7.4%	14.41	180966	371.4	**0.6**	**0.6**	**0**	**1**	**372.8**
240	18	5	1563.2 (1)	5.75%	12.23	102120	457.2	**0.6**	**0.6**	**0**	**1**	**457.6**
240	18	6	2.86% (0)	4.48%	10.76	81552	535.4	**0.6**	**0.7**	**0**	**1**	**536**
240	18	7	1166.1 (3)	2.4%	8	62051	**612.4**	**0.6**	**0.7**	**0**	**1**	**612.4**
240	18	8	278.2	912	5.15	22099	**683**	**0.6**	**0.6**	**0**	**1**	**683**
240	18	9	28	62.7	3.1	1499	**740.6**	**0.6**	**0.7**	**0**	**1**	**740.6**
240	18	10	14	32.9	2.21	201	**783.4**	**0.6**	**0.7**	**0**	**1**	**783.4**
240	18	11	6.1	11.9	1.76	13	**819.6**	**0.6**	**0.7**	**0**	**1**	**819.6**
240	18	12	4.9	9.7	1.41	5	**852**	**0.6**	**0.7**	**0**	**1**	**852**
240	18	13	5	9	1.29	10	**880.2**	**0.6**	**0.7**	**0**	**1**	**880.2**
240	18	14	3.4	7	1.1	2	**906.8**	**0.7**	**0.7**	**0**	**1**	**906.8**
240	18	15	2.5	4.6	0.85	**1**	**932.2**	**0.6**	**0.7**	**0**	**1**	**932.2**
240	18	16	1.8	3.4	0.45	**1**	**957.2**	**0.6**	**0.6**	**0**	**1**	**957.2**
240	18	17	0.9	2.2	0.2	**1**	**979.8**	**0.6**	**0.6**	**0**	**1**	**979.8**
240	18	18	**0.3**	**0.3**	**0**	**1**	**1000**	0.6	0.6	**0**	**1**	**1000**

Table 3.4: Computational results comparing the rescheduling formulations F^{st} and F^{sut}

Table **??** reports the comparison between formulations F^{st} and F^{sut}. Even when F^{st} provides optimal solutions in small running times until $|T| = 120$, longer times are required for some instances of $|T| = 180$ and not all instances can be solved to optimality for some instances of $|T| = 240$. Column

$F^{st}.gapLR$ shows that only instances where $|K_0| = \kappa$ can be solved with linear programming. However, according to $F^{st}.nodes$ many of the instances where solved in the root node just adding preprocessing cuts. In general among all tables, column obj shows how grows the objective function (passengers total convenience after the rescheduling) with the number of vehicles that are rescheduled. Table **??**.F^{sut} shows how clearly formulation is F^{st} outperformed by F^{sut}. We proved that F^{sut} can be solved with linear programming by using an equivalent formulation. However, F^{sut} already seems to have the integrality property since all the instances where solved in the rood node without any preprocessing cuts.

$\|T\|$	κ	$\bar{\kappa}$	$t/gap(\#)$	t^*/gap^*	$gapLR$	nod	obj	$t/gap(\#)$	t^*/gap^*	$gapLR$	nod	obj	$t/gap(\#)$	t^*/gap^*	$gapLR$	nod	obj
60	4	1	3	8.6	14.55	4	**415.6**	0.2	0.7	10.77	**1**	**415.6**	**0.1**	**0.1**	**0**	**1**	**415.6**
60	4	2	6.5	26.6	3.84	12	**761.2**	**0.3**	1.2	1.21	**1**	**761.2**	0.4	**0.4**	**0**	**1**	**761.2**
60	4	3	32	77.1	3.53	522	**898**	3.9	7.6	1.56	11	**898**	**0.5**	**0.6**	**0**	**1**	**898**
60	4	4	1.1	1.2	**0**	**1**	**1000**	**0.3**	**0.3**	**0**	**1**	**1000**	0.6	0.6	**0**	**1**	**1000**
120	9	1	4.2	8.9	15.54	**1**	**199.6**	1.3	3.1	11.34	**1**	**199.6**	**0.6**	**0.6**	**0**	**1**	**199.6**
120	9	2	610	1103.6	14.7	21456	**378.8**	22.5	28.4	10.69	175	**378.8**	**0.9**	**0.9**	**0**	**1**	**378.8**
120	9	3	4% (0)	6.09%	12.36	43180	**541.2**	81.7	123.2	8.6	4138	**541.2**	**1.3**	**1.3**	**0**	**1**	**541.2**
120	9	4	1765.9 (1)	5.78%	7.46	16730	689.6	130.8	225.4	3.99	4752	**690.8**	**1.8**	**1.9**	**0**	**1**	**690.8**
120	9	5	2.18% (0)	5.66%	5	8236	793	420.3 (4)	0.75%	1.76	11179	**794.2**	**2.3**	**3**	**0**	**1**	**794.2**
120	9	6	2.39% (0)	3.95%	4.75	10199	854.8	475.3 (4)	0.67%	1.43	6276	**856.6**	**2.9**	**3.6**	**0**	**1**	**856.6**
120	9	7	2.3% (0)	2.87%	3.77	8179	908.2	504	1072.6	0.99	5345	**908.4**	**3.6**	**4.1**	**0**	**1**	**908.4**
120	9	8	1462.1 (1)	1.47%	2.25	6348	**956.4**	66.1	116.6	0.47	125	**956.4**	**4.2**	**5.2**	**0**	**1**	**956.4**
120	9	9	10.1	12.6	**0**	**1**	**1000**	**1.4**	**1.5**	**0**	**1**	**1000**	3.4	3.5	**0**	**1**	**1000**
180	14	1	24.9	38	19.93	162	**128.8**	5.9	12	15.02	6	**128.8**	**1.2**	**1.2**	**0**	**1**	**128.8**
180	14	2	1622.7 (1)	8.21%	18.01	34985	**251.8**	50.5	58.6	13.39	1778	**251.8**	**1.3**	**1.4**	**0**	**1**	**251.8**
180	14	3	8.91% (0)	11.4%	16.48	26553	364.6	627.7 (4)	2.05%	11.7	33292	**365.6**	**2.1**	**2.4**	**0**	**1**	**365.6**
180	14	4	9.46% (0)	11.83%	14.95	9648	470.2	1753.1 (1)	5.61%	10.4	57412	**471.2**	**3.3**	**3.8**	**0**	**1**	**471.2**
180	14	5	8.91% (0)	11.4%	13.67	6144	566.4	5.11% (0)	6.65%	8.95	51530	**569.8**	**4.6**	**5.1**	**0**	**1**	**569.8**
180	14	6	7.91% (0)	10.61%	11.11	4090	655.2	3.78% (0)	5.61%	6.03	18722	663.4	**5.8**	**6.3**	**0**	**1**	**663.6**
180	14	7	7.39% (0)	8.6%	9.81	3260	721.6	1.96% (0)	3.43%	3.27	6701	742.2	**7.5**	**10.9**	**0**	**1**	**742.6**
180	14	8	9.3% (0)	16.83%	12.02	1476	754.2	1424.4 (2)	2.15%	1.94	3405	**798.6**	**8.3**	**10.1**	**0**	**1**	**798.6**
180	14	9	6.14% (0)	9.33%	7.57	950	822.2	1745.7 (1)	1.27%	1.44	3294	842.4	**10.1**	**12**	**0**	**1**	**842.6**
180	14	10	5.48% (0)	6.79%	6.76	682	861.2	1461.4 (1)	1.42%	1.38	1822	878.4	**12.3**	**15**	**0**	**1**	**879.4**
180	14	11	3.96% (0)	4.72%	5.03	1051	902.6	1460.1 (1)	1.29%	1.17	1696	911.6	**17.7**	**21.3**	**0**	**1**	**911.8**
180	14	12	3.26% (0)	5.03%	4.09	1188	933.4	1441.7 (1)	0.99%	0.81	2100	943.2	**19**	**24**	**0**	**1**	**943.4**
180	14	13	1.27% (0)	1.68%	1.92	1225	971.4	1161.5 (2)	0.32%	0.41	5263	**972.6**	**19**	**21.4**	**0**	**1**	**972.6**
180	14	14	54.8	66.9	**0**	**1**	**1000**	**3.6**	**3.7**	**0**	**1**	**1000**	10.3	10.5	**0**	**1**	**1000**
240	18	1	43.1	60.1	21.51	267	**100**	10.1	13.6	16.31	18	**100**	**1.3**	**1.9**	**0**	**1**	**100**
240	18	2	8.38% (0)	14.17%	21.04	34656	195.8	98	235.6	16.06	1951	**196**	**1.9**	**2.4**	**0**	**1**	**196**
240	18	3	12.76% (0)	19.23%	21.51	12929	283.2	856.6 (4)	7.38%	15.52	76627	**286.2**	**4**	**4.5**	**0**	**1**	**286.2**
240	18	4	13.22% (0)	15.43%	20.3	5288	367.8	7.77% (0)	11.38%	13.97	63896	**372.8**	**5.1**	**6.9**	**0**	**1**	**372.8**
240	18	5	14.63% (0)	18.12%	21.16	2672	440.2	8.07% (0)	10.45%	12.23	35816	457.2	**8.8**	**10.7**	**0**	**1**	**457.6**
240	18	6	14.28% (0)	17.88%	19.27	1675	515.6	8.59% (0)	10.63%	10.76	14374	535.4	**14.5**	**16.4**	**0**	**1**	**536**
240	18	7	13.68% (0)	22.94%	18.57	561	578.8	7.52% (0)	9.26%	8.78	5732	608	**19.6**	**20.8**	**0**	**1**	**612.4**
240	18	8	12.81% (0)	19.35%	17.01	350	636.6	5.91% (0)	8.86%	6.65	2304	673.6	**29.1**	**31.9**	**0**	**1**	**683**
240	18	9	14.23% (0)	23.47%	18.61	236	668.8	5.06% (0)	7.1%	5.65	1945	722.8	**32.8**	**34.9**	**0**	**1**	**740.6**
240	18	10	10.31% (0)	17.09%	12.9	164	736	4.16% (0)	5.5%	4.51	1738	766.2	**38.2**	**45**	**0**	**1**	**783.4**
240	18	11	8.09% (0)	11.22%	9.92	139	786.4	2.87% (0)	3.82%	3.1	830	809	**45.2**	**49.6**	**0**	**1**	**819.6**
240	18	12	10% (0)	13.22%	12.18	81	797.8	2.7% (0)	4.66%	2.88	633	840	**47.8**	**58.6**	**0**	**1**	**852**
240	18	13	9.09% (0)	12.32%	11.01	87	830.4	1.26% (0)	1.91%	1.35	689	879.6	**55.1**	**61.3**	**0**	**1**	**880.2**
240	18	14	7.08% (0)	11.91%	8.64	70	869.8	1.31% (0)	1.75%	1.41	643	904	**61.9**	**67.2**	**0**	**1**	**906.8**
240	18	15	5.14% (0)	8.19%	6.36	67	907	1509.7 (1)	1.24%	0.98	468	931	**66.5**	**72.1**	**0**	**1**	**932.2**
240	18	16	4.16% (0)	6.06%	5	162	934	1449.9 (1)	0.8%	0.57	1864	956	**66.5**	**75.8**	**0**	**1**	**957.2**
240	18	17	7.64% (0)	32.39%	11.06	115	912.8	367.2	665.2	0.2	10	**979.8**	**65.9**	**72.1**	**0**	**1**	**979.8**
240	18	18	164.2	202.2	**0**	**1**	**1000**	**7.8**	**8.8**	**0**	**1**	**1000**	20.5	21.2	**0**	**1**	**1000**
			F^{sutk} as in Mesa et al. (2013)					F^{stk}					F^{sutk}				

Table 3.5: Computational results comparing the 4-index rescheduling formulation F^{sutk} as presented in (Mesa et al., 2013) with the current models F^{st} and F^{sut}

From the results in Table **??** we observe first that F^{sutk} provides worse running times and gaps that the previous analyzed F^{st} and F^{sut} formulations. As mentioned in Section 3.4.4 working with those formulations do not provide any advantage unless extra side constraints related to the features of different vehicles are included. Otherwise, we just enlarge the size of the formulation as well as the set of feasible symmetric solutions. On the contrary, although F^{sutk} is still solved as a linear program, the running times of the larger instances ($|T| = 240$) show an increase on de difficulty for solving these problems. Finally, our implementation of F^{sutk} as described in (Mesa et al., 2013) is clearly outperformed by any of the other formulations in terms of both running time and gaps.

Summing up, formulation F^{sut} can be used to solve large instances in running times lower than 1 second which fulfills the requirements for an efficient on-line rescheduling. In addition, we are also able to cope with a bigger formulation F^{sutk} in reasonable times (1 minute in the worst case). This is an improvement with respect to the results of Mesa et al. (2013). Regarding to formulations F^{st} and F^{stk}, even when the provided results are not competitive with F^{sut} and/or F^{sutk}, they do not require to use index u and this might be determinant if other side constraints are added to the formulation. In this latter case, we refer to the heuristic approaches developed in Mesa et al. (2013).

3.5.2 Application to real data

We have tested the presented methodology in a real instance of the commuter train systems of Madrid. Figure 3.3 shows a section of Line C4 (Parla [s=1] – Getafe Sector 3 [s=2] – Getafe Centro [s=3] – Las Margaritas Universidad [s=4] – Villaverde Alto [s=5] – Villaverde Bajo [s=6] – Atocha [s=7]) that we will consider in our study. Table 3.6 shows departure times at stations of all trains that complete the itinerary Parla-Atocha in the time period [6:00,9:00], as well as the number of passengers boarding trains in each station.

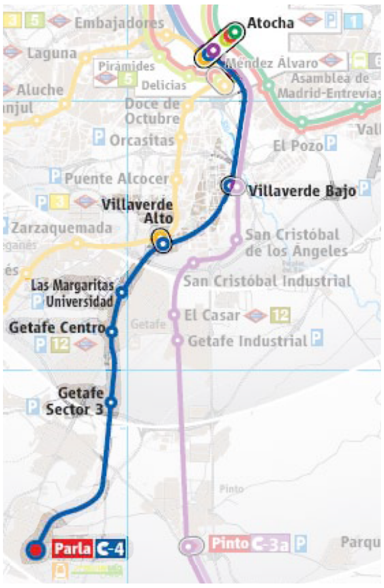


Figure 3.3: Line C4 (Parla-Atocha)

	Stations													
	$s=1$		$s=2$		$s=3$		$s=4$		$s=5$		$s=6$		$s=7$	
k	t'	$a_{st'}$	t'	$a_{st'}$	t'	$a_{st'}$	t'	$a_{st'}$	t'	$a_{st'}$	t'	$a_{st'}$	t'	$a_{st'}$
1	6:04	335	6:10	1	6:13	44	6:15	7	6:18	44	6:21	46	6:31	147
2	6:12	177	6:18	5	6:21	113	6:24	48	6:26	124	6:29	81	6:38	302
3	6:16	307	6:22	1	6:25	35	6:28	29	6:30	64	6:34	58	6:42	123
4	6:22	55	6:28	8	6:31	138	6:33	54	6:36	163	6:39	86	6:48	234
5	6:28	429	6:34	10	6:36	145	6:39	62	6:42	173	6:44	119	6:54	349
6	6:34	511	6:40	4	6:42	102	6:44	26	6:46	153	6:50	115	7:00	571
7	6:40	484	6:46	12	6:48	151	6:50	54	6:52	119	6:57	107	7:07	129
8	6:46	491	6:52	10	6:54	166	6:56	70	7:00	157	7:04	160	7:12	184
9	6:54	414	7:00	24	7:03	254	7:05	119	7:08	158	7:12	185	7:20	490
10	7:01	476	7:07	17	7:10	195	7:12	88	7:15	106	7:18	145	7:26	514
11	7:09	421	7:15	33	7:18	260	7:20	111	7:24	146	7:27	209	7:35	491
12	7:16	550	7:22	38	7:25	218	7:27	158	7:30	123	7:34	298	7:43	574
13	7:22	414	7:28	36	7:31	247	7:34	119	7:38	136	7:41	243	7:48	451
14	7:28	421	7:34	26	7:37	127	7:41	104	7:44	97	7:47	154	7:54	276
15	7:34	386	7:40	31	7:42	145	7:45	113	7:48	103	7:53	144	8:00	424
16	7:40	384	7:46	47	7:48	171	7:50	108	7:54	99	7:58	180	8:06	284
17	7:46	323	7:52	31	7:54	202	7:57	134	8:00	128	8:04	231	8:12	647
18	7:52	408	7:58	19	8:01	190	8:02	77	8:05	84	8:09	192	8:18	446
19	7:58	441	8:03	49	8:06	210	8:09	91	8:13	119	8:15	223	8:24	335
20	8:04	165	8:10	47	8:13	229	8:15	110	8:19	126	8:23	259	8:30	338
21	8:11	347	8:15	44	8:18	225	8:21	134	8:25	98	8:28	165	8:36	302
22	8:16	336	8:22	38	8:25	294	8:28	112	8:30	79	8:33	158	8:42	271
23	8:22	317	8:28	33	8:31	230	8:34	119	8:36	119	8:40	156	8:48	410
24	8:28	335	8:34	36	8:37	119	8:39	91	8:41	61	8:45	144	8:54	364
25	8:34	265	8:40	13	8:43	117	8:45	66	8:47	43	8:52	153	9:00	381

Table 3.6: Timetables and boarding in Line C4

Figure **??** shows in blue lines the 25 initial timetables provided by the company, and by bold red lines, the optimal rescheduling for only 9 trains according to the developed model. If the myopic selection of the 9 most efficient timetables were decided, the number of served passengers would be 15629 (see Mesa et al., 2013 for further details on data generation). An improvement of 20.9% can be reached if the rescheduling is performed by applying the model (18901 passengers) to determine 9 optimal line runs.

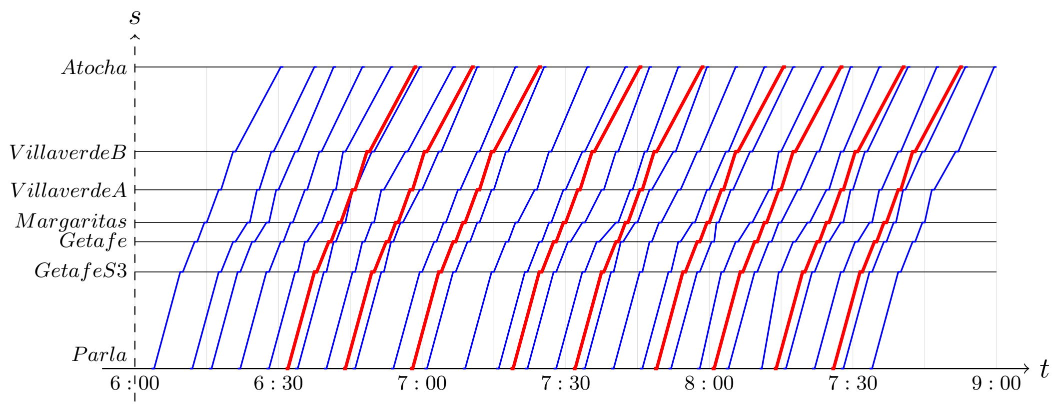


Figure 3.4: Timetables (in blue) and 9 rescheduled timetables (bold red)

3.6 Conclusions

In this chapter we have presented a modeling approach for solving the rescheduling problem in a transit line that has suffered a fleet size reduction. We have described a demand pattern to reflect the passengers' behaviour when some vehicle services are delayed or cancelled. This inconvenience function has been used to derive a rescheduling framework coming from a timetabling formulation. We have shown that the problem can be solved rapidly by using a constrained max-cost-flow problem whose coefficient matrix we prove is totally unimodular. We have tested the different formulations over a testbed of random instances and the results show that (1) on-line rescheduling can be efficiently done by using the proposed models, (2) previous approaches in the literature are outperformed and (3) our approach can be applied to real scenarios as it is the case of the commuter train system of Madrid.

Bibliography

Burdett, R. and Kozan, E. (2009) "Techniques for inserting additional trains into existing timetables". *Transportation Research Part B: Methodological*, vol. 43(8–9): 821–836.

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L. and Wagenaar, J. (2014) "An overview of recovery models and algorithms for real-time railway rescheduling". *Transportation Research Part B: Methodological*, vol. 63(0): 15–37.

Church, R. L. and ReVelle, C. (1974) "The maximal covering location problem". *Papers of the Regional Science Association*, vol. 32: 101–118.

Kumazawa, K., Hara, K. and Koseki, T. (2010) *A novel train rescheduling algorithm for correcting disrupted train operations in a dense urban environment*. Wit Press/Computational Mechanics.

Mesa, J. A., Ortega, F., Ángel Pozo, M. and Puerto, J. (2014a) "Rescheduling railway timetables in presence of passenger transfers between lines within a transportation network". In J. F. Sousa and R. Rossi, eds., *Computer-based Modelling and Optimization in Transportation*, vol. 262 of *Advances in Intelligent Systems and Computing*, pp. 347–360. Springer International Publishing Switzerland.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2009a) "Determining railway timetables according to users' desires and fleet size constraints". In *Proceedings of the XXIII EURO (European Conference on Operational Research)*, vol. 1, pp. 151–151.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2009b) "Effective allocation of fleet frequencies by reducing intermediate stops and short turning in transit systems". *Lecture Notes in Computer Science*, vol. 5868(8): 293–309.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2013) "A geometric model for an effective rescheduling after reducing service in public transportation systems". *Computers & Operations Research*, vol. 40(3): 737–746.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2014b) "Locating optimal timetables and vehicle schedules in a transit line". *Annals of Operations Research*, vol. 222: 439–455.

Nakamura, T., Hirai, C. and Nishioka, Y. (2011) "A practical train rescheduling algorithm using three predetermined factors". In *Proceedings of the 4th International Seminar on Railway Operations Modelling and Analysis*.

Sato, K., Tamura, K. and Tomii, N. (2013) "A mip-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers". *Journal of Rail Transport Planning & Management*, vol. 3(3): 38–53.

van Exel, N. A. and Rietveld, P. (2001) "Public transport strikes and traveller behaviour". *Transport Policy*, vol. 8(4): 237–246.

Vieira, G., Herrmann, J. and Lin, E. (2003) "Rescheduling manufacturing systems: A framework of startegies, policies, and methods". *Journal of Scheduling*, vol. 6: 39–62.

Chapter 4

The multi-criteria p-facility median location problem on networks

ABSTRACT

In this chapter we discuss the multi-criteria p-facility median location problem on networks with positive and negative weights. We assume that the demand is located at the nodes and can be different for each criterion under consideration. The goal is to obtain the set of Pareto-optimal locations in the graph and the corresponding set of non-dominated objective values. To that end, we first characterize the linearity domains of the distance functions on the graph and compute the image of each linearity domain in the objective space. The lower envelope of a transformation of all these images then gives us the set of all non-dominated points in the objective space and its preimage corresponds to the set of all Pareto-optimal solutions on the graph. For the bicriteria 2-facility case we present a low order polynomial time algorithm. Also for the general case we propose an efficient algorithm, which is polynomial if the number of facilities and criteria is fixed.

Keywords: Network Location, multicriteria optimization, p-facility location.

4.1 Introduction

Many real-world applications are concerned with finding an optimal location for one or more new facilities on a network (road network, power grid, etc.) minimizing a function of the distances between these facilities and a given set of existing facilities (clients, demand points), where the latter typically coincide with vertices. For a recent book on location theory the reader is referred to Nickel and Puerto (2005) and references therein. Since the first seminal paper by Hakimi (1964), an ever growing number of results have been published in this field.

The majority of research focuses on the minimization of a single objective function that is increasing with distance. However, in the process of locating a new facility usually more than one decision maker is involved. This is due to the fact that often the cost incurred with the decision is relatively high. Furthermore, different decision makers may (or will) have different (conflicting) objectives. In other situations, different scenarios must be compared due to uncertainty of data or still undecided parameters of the model. One way to deal with these situations is to apply scenario analysis. Another way of reflecting uncertainty in the parameters is to consider different replications of the objective function. Hence, there exists a large number of real-world problems which can only be modelled suitably through a multicriteria approach, especially when locating public facilities.

An additional difficulty is that we are usually dealing with conflicting criteria and a single optimal solution does not always exist (which would be an optimal solution for each of the criteria). Therefore, an alternative solution concept has to be used. One possibility is to determine the set of non-dominated solutions. That is, solutions such that there exists no other solution which is at least as good for all decision makers and strictly better for at least one of them. These solutions are often called Pareto-optimal. For an overview on multicriteria location problems the reader is referred to Nickel et al. (2005).

In contrast to the practical needs described above, network location research involving multiple criteria has received little attention, especially when it comes to multiple facilities. In this chapter, we consider the p-facility median location problem with several objective functions. Hereby, each objective function is representing the goal of one decision maker and the aim is to locate p facilities in order to minimize the total weighted distance from the clients to their closest facility. The weights assigned to clients vary from one decision maker to another, yielding different objective functions. It might even happen that one of the facilities is desirable for some decision makers and, at the same time, undesirable for others. Undesirable facilities are usually modelled using negative weights. See Eiselt and Laporte (1995) for more details on these problems. Before we discuss the literature, we present a practical example for this model. Suppose we want to locate two garbage dumps and we have a set of residential and recreational areas and a set of industrial sites where garbage has to be collected. There are two decision makers involved: the "Business economist" who has to keep the costs in check and the "Politician" who is concerned about the nuisance of the garbage dumps and the garbage trucks on the population. The business economist wants the dumps to be close to all sites to minimize travel times and costs. To that end he associates positive weights with the residential and industrial areas that are proportional to the average number of required garbage collections. In contrast to that, the politician wants to minimize the nuisance of the garbage dumps and of the trucks frequenting the garbage dumps for the population. Therefore, he assigns to each site a second, negative value. The smaller the weight is, the more likely it is that the residential area is far away from the dumps and the less likely it is that trucks that are not bound for these areas are simply passing through them on their way to the dumps. Formulating this problem in mathematical terms results in a bi-criteria 2-facility location model.

There are many other applications of multicriteria multi-facility location problems. Bitran and Lawrence (1980) consider the multicriteria location of regional service offices in the expanding operating territories of a large property and liability insurer. These offices serve as first line administrative centers for sales support and claims processing. Another application of multiobjective optimization in the context of location theory can be found in Johnson (2001) that discusses a spatial decision making problem for housing mobility planning. Ehrgott and Rau (1999) present an analysis of a part of the distribution system of BASF SE, which involves the construction of warehouses at various locations. The authors evaluate 14 different scenarios and each of these scenarios is evaluated with the minimal cost solution obtained through linear programming and the resulting average delivery time at this particular solution. For more applications see Schöbel (2005), Carrano et al. (2007), and Kolokolov and Zaozerskaya (2013).

Concerning the methodological aspects of multicriteria network location problems, Hamacher et al. (1999) discuss the network 1-facility problem with median objective functions. They show that for Pareto-optimal locations on undirected networks no node dominance result can be proven. Hamacher et al. (2002) provide a polynomial time algorithm for the 1-facility problem when the objectives are both weighted median and anti-median functions. The method is generalized for any piecewise linear objective function. Zhang and Melachrinoudis (2001) develop a polynomial algorithm for the 2-criteria 1-facility network location problem maximizing the minimum weighted distance from the service facility to the nodes (maximin) and maximizing the sum of weighted distances between the service facility and the nodes (maxisum). Skriver et al. (2004) introduce two sum objectives and criteria dependent edge lengths for the 1-facility 2-criteria problem. Nickel and Puerto (2005) solve the 1-facility problem when all objective functions are ordered medians. Colebrook and Sicilia (2007a,b) provide polynomial algorithms for solving the cent-dian 1-facility location problem on networks with criteria dependent edge lengths and facilities being attractive or obnoxious.

Concerning the single criterion multi-facility location problem on networks, Kalcsics (2011) derives a finite domination set for the p-median problem with positive and negative weights. For the 2-facility case, the author presents an efficient solution procedure using planar arrangements. Based on this approach, Kalcsics et al. (2015) solve the 2-facility case for different equity measures.

Many of the previous papers study the problem on trees as a particular case of generalized networks. The first work dealing with several objectives and facilities is provided by Tansel et al. (1982) who develop an algorithm for finding the efficient frontier of the biobjective multifacility minimax location problem on a tree network. This problem involves as objective functions the maximum of the weighted distances between specified pairs of new and existing facilities.

Despite its intrinsic interest as discussed above, to the best of our knowledge there are no papers discussing the multicriteria p-facility median location problem on networks and no results are known until the moment to obtain the set of Pareto-optimal solutions.

The remainder of this chapter is organized as follows. Section 4.2 introduces the notation and concepts used throughout the chapter. Section 4.3 presents some properties of the k-criteria p-facility median problem on networks. Section 4.4 is devoted to the development of a polynomial algorithm for the 2-criteria 2-facility version of the problem. A solution procedure for the general case is proposed in Section 4.5. Finally, Section 4.6 contains some conclusions and posible extensions of the analyzed problems.

4.2 Problem description and general concepts

4.2.1 Problem definition

Let $G = (V, E)$ be an undirected connected graph with node set $V = \{v_1, \dots, v_n\}$ and edge set $E = \{e_1, \dots, e_m\}$. Each edge $e \in E$ has a positive length $\ell(e)$, and is assumed to be rectifiable. Let $A(G)$ denote the continuum set of points on edges of G. We denote a point $x \in e = [u, v]$ as a pair $x = (e, t)$, where t $(0 \leq t \leq 1)$ gives the relative distance of x from node u along edge e. For the sake of readability, we identify $A(G)$ with G and $A(e)$ with e for $e \in E$. Let $k \geq 1$ be the number of criteria of the problem and define $Q = \{1, \dots, k\}$. Each vertex $v_i \in V$ has a real-valued weight $w_i^q \in \mathbb{R}$, $q \in Q$. Let $J = \{1, \dots, p\}$, where p is the number of facilities to be located. We denote by $X = (x_1, \dots, x_p)$ the vector of locations of the facilities, where $x_j \in G$, $j \in J$. (Note that in order to allow co-location, which is quite common in location problems with negative weights, we have to represent the facility locations using a vector.) In the remainder, we use the notions location vector and solution synonymously.

We denote by $d(x, y)$ the length of the shortest path connecting two points $x, y \in G$. Let $v_i \in V$ and $x = ([v_r, v_s], t) \in G$. The distance from v_i to x entering the edge $[v_r, v_s]$ through v_r (v_s) is given as $D_i^+(x) = d(v_r, x) + d(v_r, v_i)$ $(D_i^-(x) = d(v_s, x) + d(v_s, v_i))$. Hence, the length of a shortest path from v_i to x is given by $D_i(x) = \min\{D_i^+(x), D_i^-(x)\}$. As $d(v_r, x) = t \cdot \ell(e)$ and $d(v_s, x) = (1 - t) \cdot \ell(e)$, the functions $D_i^+(x)$ and $D_i^-(x)$ are linear in x and $D_i(x)$ is piecewise linear and concave in x, cf. Drezner (1995). The distance from v_i to its closest facility is finally defined as $D_i(X) = \min_{j \in J} D_i(x_j) = \min_{j \in J}\{D_i^+(x_j), D_i^-(x_j)\}$. In the following, we call the functions $D_i^{+/-}(x)$ and $D_i(X)$ distance functions of node v_i. Moreover, we say that $D_i^a(x_j)$, $a \in \{+, -\}$, is active for X, if $D_i^a(x_j) = D_i(X)$.

We consider the objective function $F(X) = (F^1(X), \dots, F^{|Q|}(X))$, where each $F^q(X)$, $q \in Q$, is a median function defined as:

$$F^q(X) = \sum_{i \in V} w_i^q \, D_i(X) \, .$$

We assume the usual definition of Pareto-optimality or efficiency (Ehrgott (2005)). That is, a solution X is called efficient or Pareto-optimal, if there exists no solution X' which is at least as good as X with respect to all objective function values and strictly better for at least one value, i.e., $\nexists X' : F_q(X') \leq F_q(X), \forall\, q \in Q$, and $\exists q \in Q : F_q(X') < F_q(X)$. If X is Pareto-optimal, $F(X) \in \mathbb{R}^k$ will be called a non-dominated point. If $F_q(X) \leq F_q(X') \;\forall\, q \in Q$ and $\exists q \in Q : F_q(X) < F_q(X')$ we say X dominates X' in the decision space and $F(X)$ dominates $F(X')$ in the objective space.

The k-criteria p-facility median location problem on networks, denoted by (k, p)-MLPN, is now defined as the problem of determining the set of all Pareto-optimal solutions on the graph:

$$\underset{X \in G \times \overset{p}{\dots} \times G}{\text{v-min}} \; F(X) \, , \tag{4.1}$$

where v-min stands for vector minimization. We denote by $\tilde{X}$ the set of all Pareto-optimal solutions of

(4.1). As mentioned in the introduction, we are interested in obtaining a description of the complete sets of Pareto-optimal solutions (in the decision space) and the non-dominated points (in the objective space). Hereby, the set of Pareto-optimal solutions comprises all alternative location vectors for the p facilities that are suitable candidates to choose from, because no other point can give rise to objective values that dominate them component-wise.

Let $h = (e_{h_1}, \ldots, e_{h_p})$ be a p-tuple of not necessarily distinct edges, where $e_{h_j} \in E$, $j \in J$. Then, the (k,p)-MLPN can be equivalently formulated as:

$$\text{v-min}\{F(X) \mid X \in e_{h_1} \times \ldots \times e_{h_p},\, h \in E \times \ldots \times E\}\,.$$

Note that because of symmetry it is sufficient to consider only p-tuples h for which $h_1 \leq \cdots \leq h_p$.

4.2.2 General concepts

Let $h = (e_{h_1}, \ldots, e_{h_p})$ be a p-tuple of edges and $X \in e_{h_1} \times \ldots \times e_{h_p}$ with $x_j = (e_{h_j}, t_j)$, $0 \leq t_j \leq 1$. In the following, we derive a subdivision of $e_{h_1} \times \ldots \times e_{h_p}$ into maximal subsets such that the distance function of each node is linear over such a subset, i.e., each node is allocated to the same facility for all location vectors in the subset and each node reaches its closest facility via the same vertex of the edge that contains this facility. This subdivision will be a building block of our solution approach.

Let $v_i \in V$. As the functions $D_i^+(x_j)$ and $D_i^-(x_j)$ are linear for $x_j \in e_{h_j}$, the distance functions $D_i(X)$ are piecewise linear and concave for $X \in e_{h_1} \times \ldots \times e_{h_p}$. Moreover, a breakpoint of $D_i(X)$ occurs if

- there are either two distinct facilities x_j and $x_{j'}$ at the same closest distance from v_i, i.e., $D_i(X) = D_i^a(x_j) = D_i^{a'}(x_{j'})$ for $a, a' \in \{+,-\}$, or if
- the shortest paths from v_i to its closest facility $x_j = ([v_r, v_s], t_j)$ via v_r and, respectively, v_s have the same length, i.e., $D_i(X) = D_i^+(x_j) = D_i^-(x_j)$.

It is noteworthy that the breakpoints of $D_i(X)$ for any $v_i \in V$ occur only for active functions $D_i^a(\cdot)$. See Example 4.1 for an illustration.

Example 4.1- Let $p = k = 2$ and consider the graph depicted in Figure 4.1. The node weights $w_i = (w_i^1, w_i^2)$ and the edge lengths are shown in the figure.

Consider the pair of edges $h = ([v_2, v_3], [v_4, v_5])$. In Figure 4.2 we depict the resulting sets of breakpoints of the distance functions over $[v_2, v_3] \times [v_4, v_5]$ (bold lines). The thin dashed lines indicate sets of intersection points between pairs of distance functions $D_i^a(\cdot)$ where at least one of the functions is not active.

The breakpoints for all other edge pairs are depicted in Appendix 1.

To derive the desired subdivision, we identify each edge of the network with the unit interval $[0,1]$. Hence, the cartesian product $e_{h_1} \times \ldots \times e_{h_p}$ of the edges of h corresponds to the unit hypercube $[0,1]^p$.

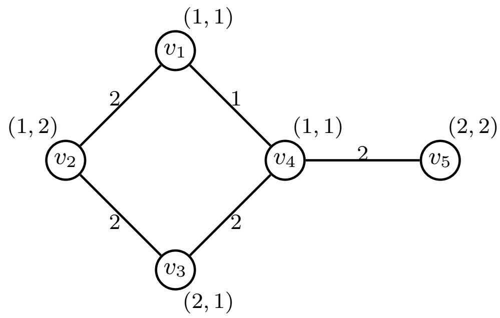


Figure 4.1: Network with node weights (in brackets) and edge lengths (Example 4.1).

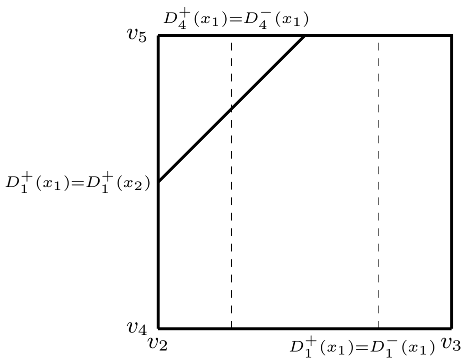


Figure 4.2: Breakpoints of the distance functions $D_i(X)$ for the pair of edges $h = ([v_2, v_3], [v_4, v_5])$.

□

For the ease of notation, we identify x_j with t_j in the remainder. The sets of location vectors that fulfill the breakpoint conditions $D(X) = D_i^a(x_j) = D_i^{a'}(x_{j'})$ and $D(X) = D_i^+(x_j) = D_i^-(x_j)$ define hyperplanes in $[0,1]^p$. The set of all these hyperplanes induces a subdivision of the hypercube into subsets such that each distance function $D_i(X)$ is linear over each subset of this subdivision. Such a subdivision is also called an arrangement and the subsets are called cells, see de Berg et al. (2008). As these hyperplanes resemble the breakpoints, each cell of the subdivision is maximal in the sense that all distance functions $D_i(X)$, $v_i \in V$, are linear over the cell. As the subdivision is induced by hyperplanes, all cells are convex polygons. For more details see Kalcsics (2011). In the following, we denote by C_h the set of all cells of the subdivision for h. Moreover, for a set $D \subseteq \mathbb{R}^n$, $ch(D)$ denotes the convex hull of D, $ext(D)$ the set of extreme points of D, and $|D|$ the cardinality of D.

Example 4.1 (cont.). Figure 4.2 shows the subdivision of $[0,1]^2$ into cells induced by the breakpoints for the edge pair h. In the following, we identify a solution $X = (([v_2, v_3], t_1), ([v_4, v_5], t_2))$ on the graph with the corresponding point $x = (t_1, t_2)$ on the unit square. Then, the two cells $\mathcal{C}^1$ and $\mathcal{C}^2$ of the subdivision are given by $\mathcal{C}^1 = ch(\{(0,0),(1,0),(1,1),(0.5,1),(0,0.5)\})$ and $\mathcal{C}^2 = ch(\{(0,0.5),(0.5,1),(0,1)\})$.

For location vector X in the relative interior of a cell we either have $D_i(x_1) < D_i(x_2)$ or $D_i(x_1) > D_i(x_2)$, i.e., each node will be served by the same facility x_1 or x_2. Moreover, for each node v_i the shortest path from the node to its closest facility x_j will always pass through the same endpoint of the edge containing the facility, i.e., we either have $D_i^+(x_j) < D_i^-(x_j)$ or $D_i^+(x_j) > D_i^-(x_j)$.

4.3 General properties for the (k, p)-MLPN

To determine the set of Pareto-optimal solutions in the graph, we have to compute all non-dominated points of the set $\{F(X) \mid X \in G \times \overset{p}{\ldots} \times G\}$ in the objective space. To that end, using the subdivision introduced in the previous section for a given p-tuple h of edges, it will be necessary to compute in a first step the images of all cells of this subdivision. Given these images for all p-tuples h, we are then able to derive the set of non-dominated points. In a last step, we have to identify the set of location vectors on the graph whose image corresponds to the non-dominated points. These location vectors then comprise the set of Pareto-optimal solutions of our problem. In this section we discus how to compute images of cells and preimages of sets of points in the objective space. Moreover, we present some properties of the objective function of the (k, p)-MLPN. The determination of the set of non-dominated points is described in the next sections.

Let $h = (e_{h_1}, \ldots, e_{h_p})$ be a p-tuple of edges, C_h the subdivision of $[0, 1]^p$ into cells, and $\mathcal{C}$ be a cell in C_h. Recall that $F(X) = (F^1(X), \ldots, F^k(X))$ is a mapping from $G \times \overset{p}{\ldots} \times G$ to $\mathbb{R}^k$. We first show how to compute images of cells.

Lemma 4.1- (Image of a cell) The function F is an affine mapping over $\mathcal{C} \in C_h$, i.e., $F : \mathcal{C} \to \mathbb{R}^q$, $F(X) = At + b$, $A \in \mathbb{R}^{k \times p}$, $b \in \mathbb{R}^k$, and $t \in [0, 1]^p$. Moreover, the image $F(\mathcal{C})$ of the cell has dimension $rank(A)$ with $0 \le rank(A) \le \min\{p, k\}$ and can be represented in the objective space as the convex hull of the images of the extreme points of $\mathcal{C}$.

Proof.

Let $X = ((e_{h_1}, t_1), \ldots, (e_{h_p}, t_p)) \in \mathcal{C}$. As each distance function $D_i(X)$ is linear over $\mathcal{C}$, so will be $F^q(X)$, $q \in Q$. Hence, we can write $F^q(X) = a_1^q t_1 + \cdots + a_p^q t_p + b^q$ where $a_j^q, b^q \in \mathbb{R}$, $q \in Q$. Therefore,

$$F(X) = \begin{pmatrix} F^1(X) \\ \vdots \\ F^k(X) \end{pmatrix} = \begin{pmatrix} a_1^1 t_1 + \cdots + a_p^1 t_p + b^1 \\ \vdots \\ a_1^k t_1 + \cdots + a_p^k t_p + b^k \end{pmatrix} = \begin{pmatrix} a_1^1 & \ldots & a_p^1 \\ \vdots & \vdots & \vdots \\ a_1^k & \ldots & a_p^k \end{pmatrix} \begin{pmatrix} t_1 \\ \vdots \\ t_p \end{pmatrix} + \begin{pmatrix} b^1 \\ \vdots \\ b^k \end{pmatrix} =: At + b$$

is an affine mapping. Moreover, $F(\mathcal{C})$ is a polytope of dimension $rank(A)$, with $0 \le rank(A) \le \min\{p, k\}$.

As F is an affine mapping over $\mathcal{C}$, it preserves collinearity and ratios of distances. Let $ext(\mathcal{C}) = \{v_c \mid$

$c = 1, \ldots, |ext(\mathcal{C})|\}$. Since $\mathcal{C}$ is convex, $F(\mathcal{C})$ is a convex set given by

$$\begin{aligned} F(\mathcal{C}) &= F\left(ch(\{ext(\mathcal{C})\})\right) = F\left(\left\{\sum_{c=1}^{|ext(\mathcal{C})|} \lambda_c v_c \mid \lambda_c \geq 0, \sum_{c=1}^{|ext(\mathcal{C})|} \lambda_c = 1\right\}\right) \\ &= \left\{\sum_{c=1}^{|ext(\mathcal{C})|} \lambda_c F(v_c) \mid \lambda_c \geq 0, \sum_{c=1}^{|ext(\mathcal{C})|} \lambda_c = 1\right\} = ch(\{F(v_c) \,|\, v_c \in ext(\mathcal{C})\}). \end{aligned}$$

□

If a proper subset $\mathcal{U}$ of the image $F(\mathcal{C})$ of a cell $\mathcal{C}$ belongs to the set of non-dominated points of $\{F(X) \mid X \in G \times \overset{p}{\ldots} \times G\}$ in the objective space, we have to derive the set of points of $\mathcal{C}$ whose image corresponds to $\mathcal{U}$. The next result provides a characterization of the preimage of a convex set $\mathcal{U} \subsetneq \mathcal{C}$. Its proof follows directly from the properties of affine mappings. We will see in the next sections why it is sufficient to restrict ourselves to convex sets $\mathcal{U}$.

Lemma 4.2- (Preimage of a set) Let $\mathcal{C} \in C_h$ be a cell and $\mathcal{U}$ be a convex subset of $F(\mathcal{C})$ with extreme points $z_1, \ldots, z_\vartheta$, $\vartheta \geq 1$. The preimage $F^{-1}(\mathcal{U})$ of $\mathcal{U}$ is given by

$$F^{-1}(\mathcal{U}) = ch(\{t \in [0,1]^p \mid z_c = At + b \text{ for some } c \in \{1, \ldots, \vartheta\}\}.$$

In this way, F^{-1} is well defined.

Remark 4.1- Note that $F^{-1}(\mathcal{U})$ depends on the cell $\mathcal{C}$. Therefore, we have to store for each point $t \in \mathbb{R}^k$ in the objective space the cell(s) who “generated” this point, i.e., to whose image $F(\mathcal{C})$ the point t belongs to.

The next example illustrates the computation of images and preimages.

Example 4.1 (cont.). Consider again the graph depicted in Figure 4.1, and the edge pair $h = (e_{h_1} = [v_2, v_3], e_{h_2} = [v_1, v_4])$. The subdivision C_h contains a single cell that coincides with the whole unit square, i.e., $C_h = \{[0,1]^2\}$. Let $X = (x_1, x_2)$ with $x_1 = (e_{h_1}, t_1)$ and $x_2 = (e_{h_2}, t_2)$.

1. Using the weights $w_1 = (3,3)$ and $w_2 = (2,1)$ for nodes v_1 and v_2 instead of the ones depicted in Figure 4.1, we obtain

$$\begin{aligned} F^1(X) &= 4t_1 + 4(1-t_1) + 3t_2 + (1-t_2) + 2(2 + (1-t_2)) = 11 \\ F^2(X) &= 2t_1 + 2(1-t_1) + 3t_2 + (1-t_2) + 2(2 + (1-t_2)) = 9\,. \end{aligned}$$

   Hence, $F(X) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} + \begin{pmatrix} 11 \\ 9 \end{pmatrix}$. Since $rank(A) = 0$, the image $F(\mathcal{C})$ of $\mathcal{C} = [0,1]^2$ is a single point, namely $(11,9)$. Furthermore, $F^{-1}(F(\mathcal{C})) = \mathcal{C} \cap \mathbb{R}^2 = \mathcal{C}$. Figure 4.3 shows $\mathcal{C}$, its image $F(\mathcal{C})$ and the preimage of $F(\mathcal{C})$.

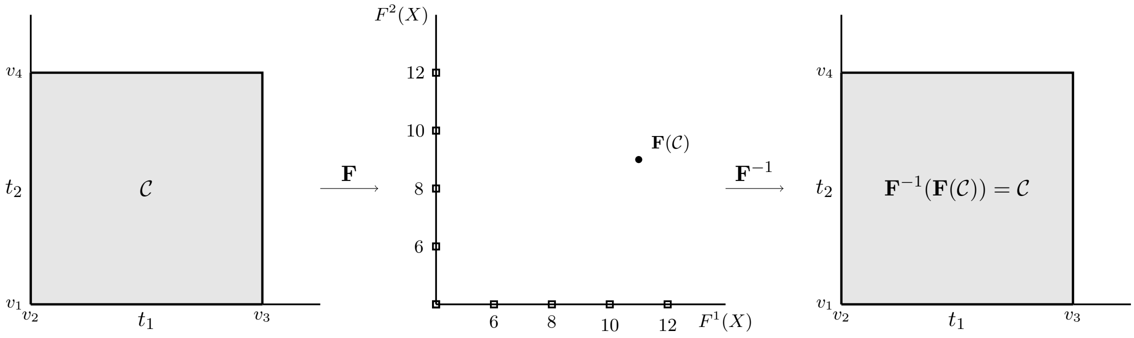


Figure 4.3: The image of a cell with $rank(A) = 0$.

2. Using now the alternative weights $w_1 = (3, 2)$ and $w_5 = (1, 2)$ for nodes v_1 and v_5, we obtain

$$\begin{aligned} F^1(X) &= 2t_1 + 4(1 - t_1) + 3t_2 + (1 - t_2) + (2 + (1 - t_2)) = 8 - 2t_1 + t_2 \\ F^2(X) &= 4t_1 + 2(1 - t_1) + 2t_2 + (1 - t_2) + 2(2 + (1 - t_2)) = 9 + 2t_1 - t_2 \,. \end{aligned}$$

Hence, $F(X) = At + b = \begin{pmatrix} -2 & 1 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} + \begin{pmatrix} 8 \\ 9 \end{pmatrix}$. Since $rank(A) = 1$, the image of $\mathcal{C}$ is now a line segment given by $ch(\{(6, 11), (9, 8)\})$. For computing the preimage, let $\mathcal{U} = ch(\{(8.5, 8.5), (7.5, 9.5)\}) \subsetneq F(\mathcal{C})$. Then,

$$\begin{aligned} F^{-1}(\mathcal{U}) &= \mathcal{C} \cap ch\left(\left\{ \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \mid \begin{pmatrix} 8.5 \\ 8.5 \end{pmatrix} = At + b \text{ or } \begin{pmatrix} 7.5 \\ 9.5 \end{pmatrix} = At + b \right\}\right) \\ &= \mathcal{C} \cap ch\left(\left\{ \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \mid 2t_1 - t_2 + 0.5 = 0 \text{ or } 2t_1 - t_2 - 0.5 = 0 \right\}\right) . \end{aligned}$$

Hence, $F^{-1}(\mathcal{U})$ is the set of all points of the square $[0, 1] \times [0, 1]$ between the two parallel lines defined by $2t_1 - t_2 + 0.5 = 0$ and $2t_1 - t_2 - 0.5 = 0$. Figure 4.4 depicts $\mathcal{C}$, its image $F(\mathcal{C})$, and the preimage of $\mathcal{U} \subseteq F(\mathcal{C})$.

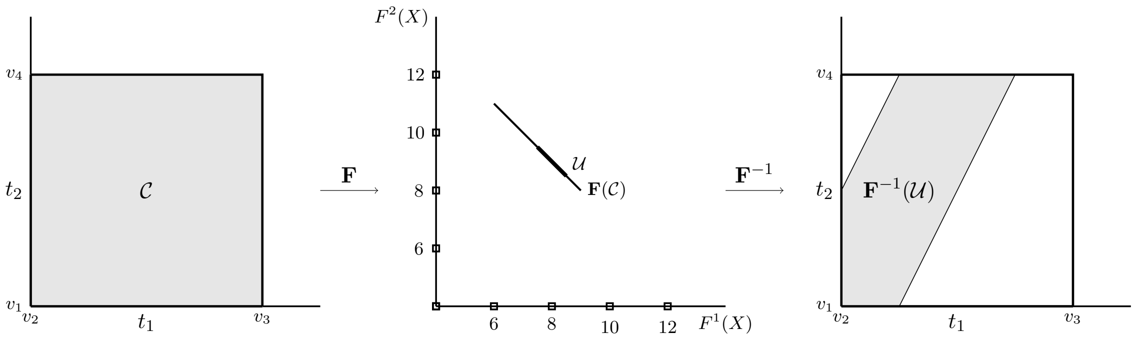


Figure 4.4: The image of a cell with $rank(A) = 1$.

3. Using the alternative weight $w_1 = (1, 2)$ for node v_1, we obtain

$$\begin{aligned} F^1(X) &= 2t_1 + 4(1-t_1) + t_2 + (1-t_2) + 2(2+(1-t_2)) = 11 - 2t_1 - 2t_2 \\ F^2(X) &= 4t_1 + 2(1-t_1) + 2t_2 + (1-t_2) + 2(2+(1-t_2)) = 9 + 2t_1 - t_2 \,. \end{aligned}$$

Hence, $F(X) = At + b = \begin{pmatrix} -2 & -2 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} + \begin{pmatrix} 11 \\ 9 \end{pmatrix}$. Since $rank(A) = 2$, the image of $\mathcal{C}$ is now a polygon with vertices $(7, 10)$, $(9, 8)$, $(11, 9)$, and $(9, 11)$. For computing the preimage, let $\mathcal{U} = ch(\{(9, 11),\ (8, 10),\ (9, 9)\}) \subsetneq F(\mathcal{C})$. Then,

$$\begin{aligned} F^{-1}(\mathcal{U}) &= \mathcal{C} \cap ch\left(\left\{ \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \mid \begin{pmatrix} 9 \\ 11 \end{pmatrix} = At + b,\ \begin{pmatrix} 8 \\ 10 \end{pmatrix} = At + b,\ \text{or}\ \begin{pmatrix} 9 \\ 9 \end{pmatrix} = At + b \right\}\right) \\ &= \mathcal{C} \cap ch(\{(1, 0),\ (5/6, 2/3), (1/3, 2/3)\})\,, \end{aligned}$$

which is again a triangle. Figure 4.5 shows $\mathcal{C}$, its image $F(\mathcal{C})$, and the preimage of $\mathcal{U} \subseteq F(\mathcal{C})$.

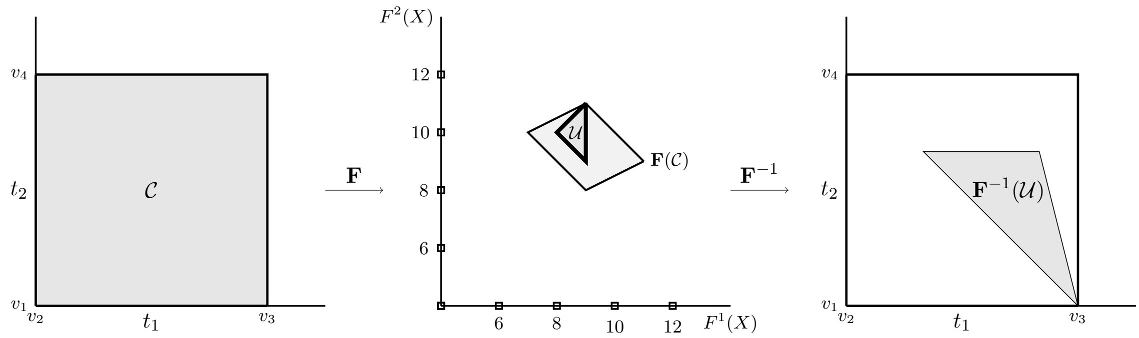


Figure 4.5: The image of a cell with $rank(A) = 2$.

4.4 A polynomial algorithm for the (2, 2)-MLPN

In this section we first discuss the (2, 2)-MLPN to explain the main ideas of our solution approach, before we turn to the general case in Section 4.5. In the following, we present the different steps of the approach. For a given pair of edges $h = (e_{h_1}, e_{h_2})$, we first compute the set C_h of cells of the subdivision of $[0, 1]^2$ into maximal domains of linearity of the distance functions $D_i(X)$. Afterwards, we compute the image $F(\mathcal{C})$ of each cell $\mathcal{C} \in C_h$. Depending on the rank of the mapping F with respect to $\mathcal{C}$, the image is either a point, a line segment, or a two-dimensional polygon, see Lemma 4.1. We store for each image a reference to the cell $\mathcal{C}$. To determine the set of non-dominated points $\tilde{Z}$ of the images of all cells in the objective space, we adapt the approach in Hamacher et al. (1999) for the single facility bi-criteria problem. The idea of their approach is to determine the set of non-dominated points in the objective space by means of the lower envelope. To facilitate that approach, they add to the rightmost point of each image a right-open horizontal halfline (as $p = 1$, all images are points or segments). Then, each point in the objective space that is not on the lower envelope will obviously be dominated by a point on the envelope. At the end, they delete all parts of the lower envelope that belong to horizontal halflines that have been added before. The remaining points then comprise the set of all non-dominated points.

Coming back to our problem, to compute the set of all non-dominated points in the objective space, we compute the image $F(\mathcal{C})$ of each cell $\mathcal{C} \in C_h$ for all edge pairs h (where it is sufficient to consider only pairs with $h_1 \leq h_2$). If the image is a point or a segment, we add it to a set $\mathcal{L}$ together with a reference to the respective cell. If $F(\mathcal{C})$ is a polygon, all interior points will be dominated by points on the boundary. Thus, we only add the bounding edges of the polygon to $\mathcal{L}$, again with a reference to the respective cell. Note that each polygon has at most eight bounding edges (Kalcsics (2011)). In view of the general case to be discussed in Section 4.5, we generalize the approach in Hamacher et al. (1999) as follows. We add to the bottommost point of each image a horizontal and a vertical halfline extending to $+\infty$. Then, we first determine the lower envelope in y-direction and afterwards the lower envelope of the remaining points in x-direction. In this way, all dominated points will be eliminated. Formally, let le_1, le_2 denote the lower envelope functions of a set with respect to the directions of the two components of the canonical basis of $\mathbb{R}^2$. Given a collection $\mathcal{L} \subset \mathbb{R}^2$ of points and segments, Procedure 4.1 summarizes the steps to obtain the set of non-dominated points in the objective space.

Procedure 4.1- (*Computing the non-dominated set of a collection* $\mathcal{L} \subset \mathbb{R}^2$)

1. For each connected component of $\mathcal{L}$ find the point (z_1, z_2) of the component which has the smallest z_2 value and augment the horizontal halfline $\{(z_1 + s, z_2) \mid s \in \mathbb{R}^+\}$ and the vertical halfline $\{(z_1, z_2 + s) \mid s \in \mathbb{R}^+\}$ to $\mathcal{L}$.

2. Compute $\tilde{Z} = (le_1 \circ le_2)(\mathcal{L})$.

The output $\tilde{Z}$ of Procedure 4.1 is a collection of segments and points. Note that by applying both lower envelope functions, all horizontal and vertical halflines added in Step 1 are deleted at the end. Each element ℓ of $\tilde{Z}$ is a point or a subset of a segment of $\mathcal{L}$ and contains a reference to the set of elements of $\mathcal{L}$ in which it is contained. Therefore, we can immediately determine the set $C(\ell)$ of all cells whose image contains ℓ. As, in turn, each element of $\mathcal{L}$ contains a reference to the cell generating this element, we can readily compute the preimage of ℓ with respect to each cell $\mathcal{C} \in C(\ell)$ using Lemma 4.2. The union of all these preimages yields the set of Pareto-optimal solutions. Algorithm 2 gives a complete description of our approach to compute the sets of non-dominated points and Pareto-optimal solutions.

Complexity analysis. In the following, we discuss the complexity of Algorithm 2. For each pair of edges, there are at most $O(n^2)$ cells in C_h (Kalcsics (2011)). Using the procedure described in Kalcsics (2011), we can compute the linear representation of F over all cells $\mathcal{C} \in C_h$ in $O(n^2)$ total time. Step 5 can be computed in constant time as each cell has at most eight extreme points (Kalcsics (2011)). Since there are at most $O(m^2)$ pairs of edges, the overall complexity of Steps 1-5 is $O(n^2m^2)$.

Concerning Step 6, the set $\mathcal{L}$ has $O(n^2m^2)$ elements. Hence, we have to add $O(n^2m^2)$ horizontal

Algorithm 2: Solution method for the (2, 2)-MLPN

Input: A graph G

Output: The sets $\tilde{Z}$ and $\tilde{X}$ of non-dominated points and, respectively, Pareto-optimal solutions

1 **for** *each pair of edges* $h = (e_{h_1}, e_{h_2}) \in E \times E, h_1 \leq h_2$ **do**
2   Compute the subdivision of $e_{h_1} \times e_{h_2}$ and the set C_h of all cells (Section 4.2.2);
3   **for** *each cell* $\mathcal{C} \in C_h$ **do**
4   Compute the linear representation of F over $\mathcal{C}$ and the image $F(\mathcal{C})$;
5   Add (the bounding edges of) $F(\mathcal{C})$ to the collection $\mathcal{L}$ and store a reference of the cell to each point/segment of the image $F(\mathcal{C})$;
6 Compute the set $\tilde{Z}$ of non-dominated points of the collection $\mathcal{L}$ using Procedure 4.1;
7 **for** *each* ℓ *of* $\tilde{Z}$ **do**
8   Determine $C(\ell)$;
9   **for** *each* $\mathcal{C} \in C(\ell)$ **do**
10   Add the set of points of the graph corresponding to $F^{-1}(\ell)$ to the set of Pareto-optimal solutions $\tilde{X}$;
11 **return** $\tilde{Z}$ *and* $\tilde{X}$

and vertical lines, which can be done in $O(n^2m^2)$ time. The lower envelope can be computed in $O(n^2m^2 \log(nm))$ and contains $O(n^2m^2\alpha(n^2m^2))$ number of elements (Hershberger, 1989), where $\alpha(\cdot)$ is the inverse of the Ackerman's function. Hence, the overall complexity for Step 6 is $O(n^2m^2 \log(nm))$.

As for the computation of preimages of elements in the set of non-dominated points, Step 8 can be done in constant time since it is part of the output of the lower envelope algorithm. Step 10 can be carried out in constant time since the preimage of a point or segment has at most eight bounding segments. Thus, the overall complexity of Steps 7-10 is equal to the number of components of $\tilde{Z}$, that is $O(n^2m^2\alpha(n^2m^2))$. With this, the overall complexity of Algorithm 2 is $O(n^2m^2 \log(nm))$.

Remark 4.2- (Speed-up improvement) If the image of a cell is a polygon it is not necessary to add all bounding segments of $F(\mathcal{C})$ to $\mathcal{L}$ since some of them will be dominated. To compute the set of all locally non-dominated bounding segments of $F(\mathcal{C})$, we first find the vertices u^1 and u^2 of $F(\mathcal{C})$ with the smallest F^1 and F^2 value, respectively. Then, starting at u^1 we add to $\mathcal{L}$ all bounding segments of $F(\mathcal{C})$ when walking from u^1 along $bd(F(\mathcal{C}))$ in clockwise direction towards u^2. Although this does not improve the worst case complexity, the actual time required to compute the lower envelope will decrease.

Example 1 (cont.).

Let $C_h = \{\mathcal{C}^1, \mathcal{C}^2\}$ be the subdivision into cells obtained in Figure 4.2 for $h = ([v_2, v_3], [v_4, v_5])$. Denoting a point $X = (([v_2, v_3], t_1), ([v_4, v_5], t_2))$ of the unit square as $X = (t_1, t_2)$, the cells $\mathcal{C}^1, \mathcal{C}^2$ can be described by $\mathcal{C}^1 = ch(\{(0, 0), (1, 0), (1, 1), (0.5, 1), (0, 0.5)\})$ and $\mathcal{C}^2 = ch(\{(0, 0.5), (0.5, 1), (0, 1)\})$. The description of $F(X)$ for $X \in [v_2, v_3] \times [v_4, v_5]$ depends on the cell under consideration and is given by:

$$F(X) = \begin{cases} \begin{pmatrix} -2 & 0 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} + \begin{pmatrix} 9 \\ 7 \end{pmatrix}, & if\ X = (t_1, t_2) \in \mathcal{C}^1; \\ \begin{pmatrix} 0 & -2 \\ 4 & -2 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} + \begin{pmatrix} 10 \\ 8 \end{pmatrix}, & if\ X = (t_1, t_2) \in \mathcal{C}^2. \end{cases}$$

Figure 4.6 shows in dashed and dotted lines the image of $\mathcal{C}^1$ and, respectively, $\mathcal{C}^2$. The set of non-dominated points $\tilde{Z}$ obtained by using Procedure 4.1 is given by:

$$\tilde{Z} = \{(6,8)\} \cup \{(9-2t_1, 7+2t_1), 0.5 < t_1 \leq 1, 0 \leq t_2 \leq 1\},$$

and is depicted in Figure 4.6 by the black segments and filled dots.

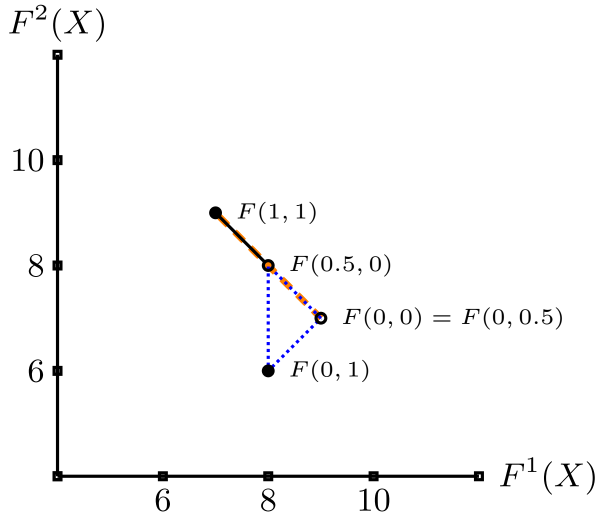


Figure 4.6: Images of cells depicted in Figure 4.2 (dashed and dotted lines) and non-dominated solutions (continuous line and filled dots).

Computing the preimages of the set $\tilde{Z}$ we obtain the following set of Pareto-optimal solutions:

$$\tilde{X} = \{(v_2, v_5)\} \cup \{(([v_2, v_3], t_1), ([v_4, v_5], t_2)) : 0.5 < t_1 \leq 1, 0 \leq t_2 \leq 1\}$$

Remark 4.3- (The $(2,2)$-MLPN on trees) If the underlying graph is a tree, the complexity of Algorithm 2 reduces by a factor of n since the number of cells of a subdivision of $[0,1]^2$ into linearity domains is at most $O(n)$, see Kalcsics (2011).

4.5 The (k,p)-MLPN

In this section we show how to extend the previous results in order to derive an algorithmic approach to solve the general problem with p facilities and k criteria, i.e., the (k,p)-MLPN. Note first that the

(k,p)-MLPN is NP-hard (if p is part of the input) because it generalizes the (1,p)-MLPN (Hakimi (1965)). Thus, there does not exist a polynomial algorithm to solve the (k,p)-MLPN (unless $\mathcal{P} = \mathcal{NP}$).

The outline of the approach for the general case is the same as for $p = k = 2$. For a given p-tuple of edges $h = (e_{h_1}, \ldots, e_{h_p})$ we first compute the set C_h of cells of the arrangement that gives us the subdivision into linearity domains (Kalcsics (2011)). For each cell $\mathcal{C} \in C_h$ we then compute the image $F(\mathcal{C})$ of $\mathcal{C}$. If $F(\mathcal{C})$ has dimension lower than k, we add it to a set $\mathcal{L}$. If $F(\mathcal{C})$ has dimension k we just add the facets of the induced polytope to $\mathcal{L}$. Moreover, we store for each element augmented to $\mathcal{L}$ a reference to its respective preimage (a cell $\mathcal{C} \in C_h$). We repeat this for all p-tuples h (where it is sufficient to consider only p-tuples with $h_1 \leq \cdots \leq h_p$). To compute the set of Pareto-optimal solutions $\tilde{X}$, we adapt Procedure 4.1 to the general case with p facilities and k criteria as follows. Again we denote by le_q the lower envelope function of a set with respect to the direction of the $q-th$ component of the canonical basis of $\mathbb{R}^k$ (see Sharir (1994) for details on the lower envelope procedure).

Procedure 4.2- (*Computing the non-dominated set $\tilde{Z}$ of a collection $\mathcal{L} \subset \mathbb{R}^k$*)
Given a collection $\mathcal{L} \subset \mathbb{R}^k$ of polytopes (of dimension lower than or equal to $k-1$) the set $\tilde{Z} \subseteq \mathcal{L}$ of non-dominated points of $\mathcal{L}$ can be obtained as follows:

1. For each element $\theta \in \mathcal{L}$ compute the convex hull of the domination cones attached to each point $c \in \theta$. Then add to $\mathcal{L}$ the set of all facets of this set.
2. Compute $(le_1 \circ \cdots \circ le_k)(\mathcal{L})$.

Algorithm **??** now gives a description of the necessary steps required to compute the set of Pareto-optimal solutions and non-dominated points.

Algorithm 3: (k,p)-MLPN

Input: A graph G, and numbers p and k
Output: The sets $\tilde{Z}$ and $\tilde{X}$ of non-dominated points and, respectively, Pareto-optimal solutions

```
for each p-tuple of edges h = (e_{h_1}, ..., e_{h_p}) ⊂ E, h_1 ≤ ··· ≤ h_p do
    Compute the set C_h of all cells of the subdivision e_{h_1} × ··· × e_{h_p} (see Section 4.2.2);
    for each cell C ∈ C_h do
        Find the linear representation of F over C;
        Add the (bounding facets of the) image F(C) to the collection L;
Compute the non-dominated subset Z̃ of the collection L by using Procedure 4.2;
for each p-tuple of edges h = (e_{h_1}, ..., e_{h_p}) ⊂ E, h_1 ≤ ··· ≤ h_p do
    for each C ∈ C_h do
        Compute ℓ = F(C) ∩ Z̃;
        Add the set of points of the graph corresponding to F^{-1}(ℓ) into the set of Pareto-optimal
        solutions X̃;
Return Z̃ and X̃;
```

In the following, we discuss the complexity of Algorithm **??**. For each p-tuple of edges, there are at most $\eta = 2np^2$ hyperplanes of the type $D_i^a(x_j) = D_i^{a'}(x_{j'}), i \in V, a, a' \in \{+,-\}, j, j' \in \{1, \ldots, p\}$.

Thus, there are $O(\eta^p)$ cells in C_h (Edelsbrunner (1987)). To compute the linear representation of F over a cell $\mathcal{C} \in C_h$, we pick an arbitrary vector $X \in \mathcal{C}$. For this X we first determine $D_i(X) = D_i^a(x_j), a \in \{+,-\}, j \in \{1,\dots,p\}$ for all $i \in V$. Afterwards, we can compute $F^q(X) = \sum_{i\in V} w_i^q D_i(X) = a_1^q t_1 + \cdots + a_p^q t_p + b^q$. Both steps require in total $O(npk)$ time and this is done only once for each cell. To compute the image of a cell we determine $F(t) = At + b$ for every extreme point $t \in ext(\mathcal{C})$, where $|ext(\mathcal{C})| = O(\eta^p)$ (Edelsbrunner (1987)), and then we plot all these extreme points (in $O(\eta^p pk)$ time). Thus, computing the image of all cells in C_h can be done in $O(\eta^{2p}pk)$. Since we have to repeat this for each p-tuple of edges, the overall effort for Steps 1-5 is $O(m^p\eta^{2p}pk)$.

Concerning Step 6, for each element $\theta \in \mathcal{L}$ we have to compute the union of the domination cones attached to each point $c \in \theta$, i.e., the set $T = \{c + \mathbb{R}^k_{\geq} \mid c \in \theta\}$. As θ is convex, this set is identical to the convex hull of the domination cones attached to each extreme point of θ, i.e., $T = ch(\{c + \mathbb{R}^k_{\geq} \mid c \in ext(\theta)\})$. To compute T for each $\theta \in \mathcal{L}$, we restrict the domination cones to $[0, M]^k$ where M is sufficiently large. Then $T = ch(\{c \cup \{a \mid a = c + Me_q, e_q = (0, \overset{q-1}{\ldots}, 0, 1, 0, \dots, 0), q \in Q\} : c \in ext(\theta)\})$ is the convex hull of $O(\eta^p + k\eta^p)$ points. Since the convex hull of τ points in $\mathbb{R}^k$ can be computed in $O(\tau^{\lfloor \frac{k}{2}+1 \rfloor})$, the convex hull of the domination cones attached to the extreme points of a cell can be computed in $O((k\eta^p)^{\lfloor \frac{k}{2}+1 \rfloor})$. Next, Step 2 in Procedure 4.2 computes the lower envelope of the set $\mathcal{L}$. Sharir (1994) shows that the complexity of the lower envelope (in one direction) in $\mathbb{R}^k$ of δ surfaces or surface patches (all algebraic of constant degree, and bounded by algebraic surfaces of constant degree) is $O(\delta^{k+\epsilon})$ for any $\epsilon > 0$ with the constant of proportionality depending on ϵ, k, s (the maximum number of intersections among any k-tuple of the given surfaces) and on the shape and degree of the surface patches of their boundaries. The number of facets of a polytope with τ extreme points in $\mathbb{R}^k$ is $O(\tau^{\lfloor \frac{k}{2}+1 \rfloor})$ (Gale, 1963). Thus, for each element $\theta \in \mathcal{L}$ the above convex hull construction generates at most $O((k\eta^p)^{\lfloor \frac{k}{2}+1 \rfloor})$ facets. Since each choice of a p-tuple of edges generates a subdivision with $O(\eta^p)$ cells, the input size of the lower envelope algorithm is $O(m^p\eta^p(k\eta^p)^{\lfloor \frac{k}{2}+1 \rfloor})$. Hence, the complexity for computing the lower envelope in each direction of the canonical basis is $O((m^p\eta^p(k\eta^p)^{\lfloor \frac{k}{2}+1 \rfloor})^{k+\epsilon})$ and the overall complexity is $O(k(m^p\eta^p(k\eta^p)^{\lfloor \frac{k}{2}+1 \rfloor})^{k+\epsilon})$. This implies that Step 6 can be computed in $O(k(m^p\eta^p(k\eta^p)^{\lfloor \frac{k}{2}+1 \rfloor})^{k+\epsilon})$. Note that, in the worst case, the output of the lower envelope algorithm contains as many elements as the cardinality of the input, that is, $O(m^p\eta^p(k\eta^p)^{\lfloor \frac{k}{2}+1 \rfloor})$.

Regarding Step 9, the number of facets (of dimension $p-1$) of a cell belonging to an arrangement of β hyperplanes in $\mathbb{R}^p$ (recall that there are $\eta = 2np^2$ hyperplanes of the type $D_i^a(x_j) = D_i^{a'}(x_{j'}), i \in V, a, a' \in \{+,-\}, j, j' \in \{1,\dots,p\}$) is bounded by $O(\beta)$ since each hyperplane can appear at most once on each cell. Then, $\ell = F(\mathcal{C}) \cap \tilde{Z}$ can be computed in $O(m^p\eta^{2p}(k\eta^p)^{\lfloor \frac{k}{2}+1 \rfloor})$ time, given that we have to process $O(\eta)$ facets of $F(\mathcal{C})$ with $O(m^p\eta^p(k\eta^p)^{\lfloor \frac{k}{2}+1 \rfloor})$ elements in the non-dominated set $\tilde{Z}$. Adding one preimage of $F^{-1}(\ell)$ requires first to compute $S = \{F^{-1}(c), c \in ext(\ell)\}$ for all extreme points c of ℓ in $O(kp\eta^p)$. Later, we compute the convex hull of S in $O((kp\eta^p)^{\lfloor \frac{p}{2}+1 \rfloor})$, and intersect the result with $[0, 1]^p$ in $O(2^p(kp\eta^p)^{\lfloor \frac{p}{2}+1 \rfloor})$ time. Thus, considering all cells in all p-tuples of edges, Steps 7-10 can be computed in $O(m^p\eta^p \max\{m^p\eta^{2p}(k\eta^p)^{\lfloor \frac{k}{2}+1 \rfloor}, 2^p(kp\eta^p)^{\lfloor \frac{p}{2}+1 \rfloor}\})$. Finally, the overall complexity of the complete algorithm is $O(\max\{k(m^p\eta^p(k\eta^p)^{\lfloor \frac{k}{2}+1 \rfloor})^{k+\epsilon}, m^p\eta^p 2^p(kp\eta^p)^{\lfloor \frac{p}{2}+1 \rfloor}\})$ for the case $k \geq 3$.

For the bicriteria problem (2,p)-MLPN it is of interest to analyze the complexity since the lower envelope can be computed efficiently using Procedure 4.1. Steps 1-5 can be computed in $O(p\eta^{2p}m^p)$ time as described in Algorithm **??**.The remaining steps are done as in Algorithm 2. As $|\mathcal{L}| = O(p\eta^{2p}m^p)$, Step 6 requires $O(p\eta^{2p}m^p\log(p\eta m))$ time (recall that the complexity of the lower envelope of τ elements in $\mathbb{R}^2$ is $O(\tau log\tau)$). The difference with respect to the complexity computed in Section 4 comes from the fact that in this case we cannot exploit that the number of extreme points of each cell is bounded by a constant (eight).

As for the computation of preimages of elements in the set of non-dominated points in the objective space, Step 8 can be done in constant time since it is part of the output of the lower envelope algorithm. Adding one preimage of $F^{-1}(\ell)$ requires first to compute $S = \{F^{-1}(c), c \in ext(\ell)\}$ for every extreme point c of ℓ in $O(p\eta^p)$. Later we compute the convex hull of S in $O((p\eta^p)^{\lfloor \frac{p}{2}+1 \rfloor})$, and intersect the result with $[0,1]^p$ in $O(2^p(p\eta^p)^{\lfloor \frac{p}{2}+1 \rfloor})$. Thus, considering all cells in all p-tuples of edges, Steps 7-10 can be computed in $O(\eta^p m^p 2^p (p\eta^p)^{\frac{p}{2}})$. Finally, the overall complexity of the complete algorithm is $O(\eta^p m^p 2^p (p\eta^p)^{\frac{p}{2}})$.

Next, we illustrate Algorithm **??** with an example of a 3-facility 3-objective problem.

Example 4.2- Let $G = (V,E)$ be the graph depicted in Figure 4.7 and let $p = k = 3$. Weights $w_i = (w_i^1, w_i^2, w_i^3)$ and edge lengths are shown in the figure. Observe that nodes v_2 and v_4 contain negative weights.

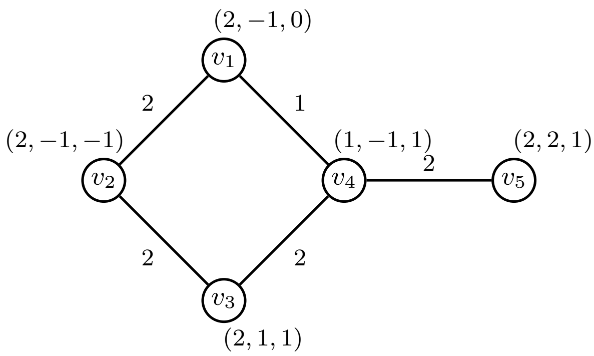


Figure 4.7: Network with node weights (in brackets) and edge lengths (Example 4.2).

As before, we identify a solution $X = ((e_1,t_1),(e_2,t_2),(e_3,t_3))$ on the graph with the corresponding point $x = (t_1,t_2,t_3)$ of the unit cube. Choosing, for example, the edges $e_1 = [v_2,v_3]$, $e_2 = [v_4,v_5]$ and $e_3 = [v_2,v_1]$, we obtain the following three hyperplanes: $D_2^+(x_1) = D_2^+(x_3)$, $D_1^-(x_3) = D_1^+(x_2)$, $D_4^-(x_3) = D_4^+(x_2)$. Figure 4.8 depicts these hyperplanes and shows the resulting subdivision into linearity domains as described in Section 4.2.2. In this case, we obtain six cells that correspond with the different possible allocations when we place a facility on each selected edge. Consider now the cell $\mathcal{C}$ delimited by the points $(1,0.5,1),(1,1,1),(1,1,0.5),(0.5,1,0.5)$ depicted in grey in Figure 4.8. In this cell v_1 and v_2 are allocated to the facility placed in e_3, v_3 to the one placed in e_1 and v_4, v_5 to the one placed in e_2. For this cell, the affine linear representation of F is given by

$$F(X) = \begin{pmatrix} F_1(X) \\ F_2(X) \\ F_3(X) \end{pmatrix} = \begin{pmatrix} -4 & -4 & -2 \\ -2 & -4 & +2 \\ -2 & -2 & -4 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} + \begin{pmatrix} 15 \\ 1 \\ 7 \end{pmatrix}.$$

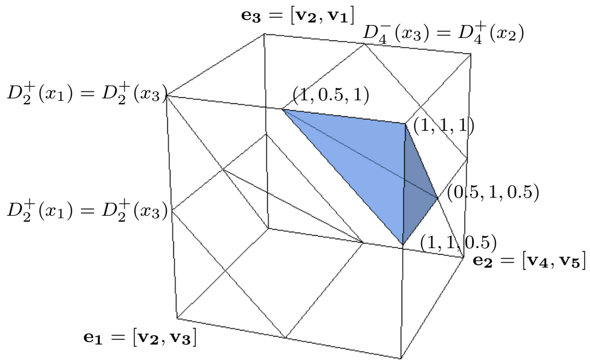


Figure 4.8: Active intersection hyperplanes and subdivision into cells for edges $e_1 = [v_2, v_3]$, $e_2 = [v_4, v_5]$ and $e_3 = [v_2, v_1]$. The cell delimited by points $(1, 0.5, 1), (1, 1, 1), (1, 1, 0.5), (0.5, 1, 0.5)$ is emphasized.

The image of $\mathcal{C}$ in the objective space is depicted in Figure 4.9.

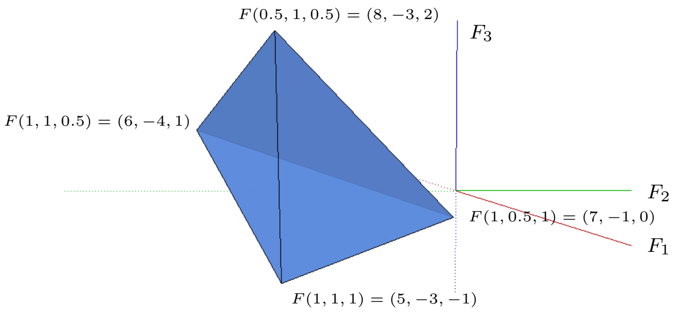


Figure 4.9: Image of the cell delimited by points $(1, 0.5, 1), (1, 1, 1), (1, 1, 0.5), (0.5, 1, 0.5)$ emphasized in Figure 4.8.

The resulting subset of non-dominated points $\tilde{Z}$ with respect to $\mathcal{C}$ is given by

$$\tilde{Z} = \{(7 - 2t_3, -5 + 2t_3, 3 - 4t_3), 0.5 \leq t_3 \leq 1\} .$$

Computing the preimages of the set $\tilde{Z}$, we obtain the following set of Pareto-optimal solutions with respect to $\mathcal{C}$:

$$\tilde{X} = \{(v_3, v_5, ([v_2, v_1], t_3)), 0.5 \leq t_3 \leq 1\} .$$

Note that two facilities are located on the vertices v_3 and v_5 and the third facility along any point of the subedge $([v_2, v_1], t_3)), 0.5 \leq t_3 \leq 1$. □

4.6 Conclusions

In this chapter we have provided a methodology to obtain a complete description of the set of Pareto-optimal solutions for the multi-criteria p-facility median location problem on networks. It is noteworthy that this chapter is the first attempt to characterize the solution set of this problem. Note that the single criteria p-facility median problem is already NP-hard and handling closest assignments makes more difficult to deal with the multifacility version.

The main tools used to obtain the set of Pareto-optimal solutions is the characterization of the linearity domains of the distance functions and the lower envelope. Hence, this analysis can be easily extended to more general objective functions as long as we can again determine these domains and their image and preimage. In this sense, an open line of research is to obtain the characterization of Pareto-optimal solutions for the case of ordered median objective functions. Recall that this function includes as particular instances most classical objectives functions used in Location Theory, as for instance the median, center, k-center and cent-dian, see Nickel and Puerto (2005) for further details.

4.7 Subdivision into linearity domains for all pairs of edges in Example 1

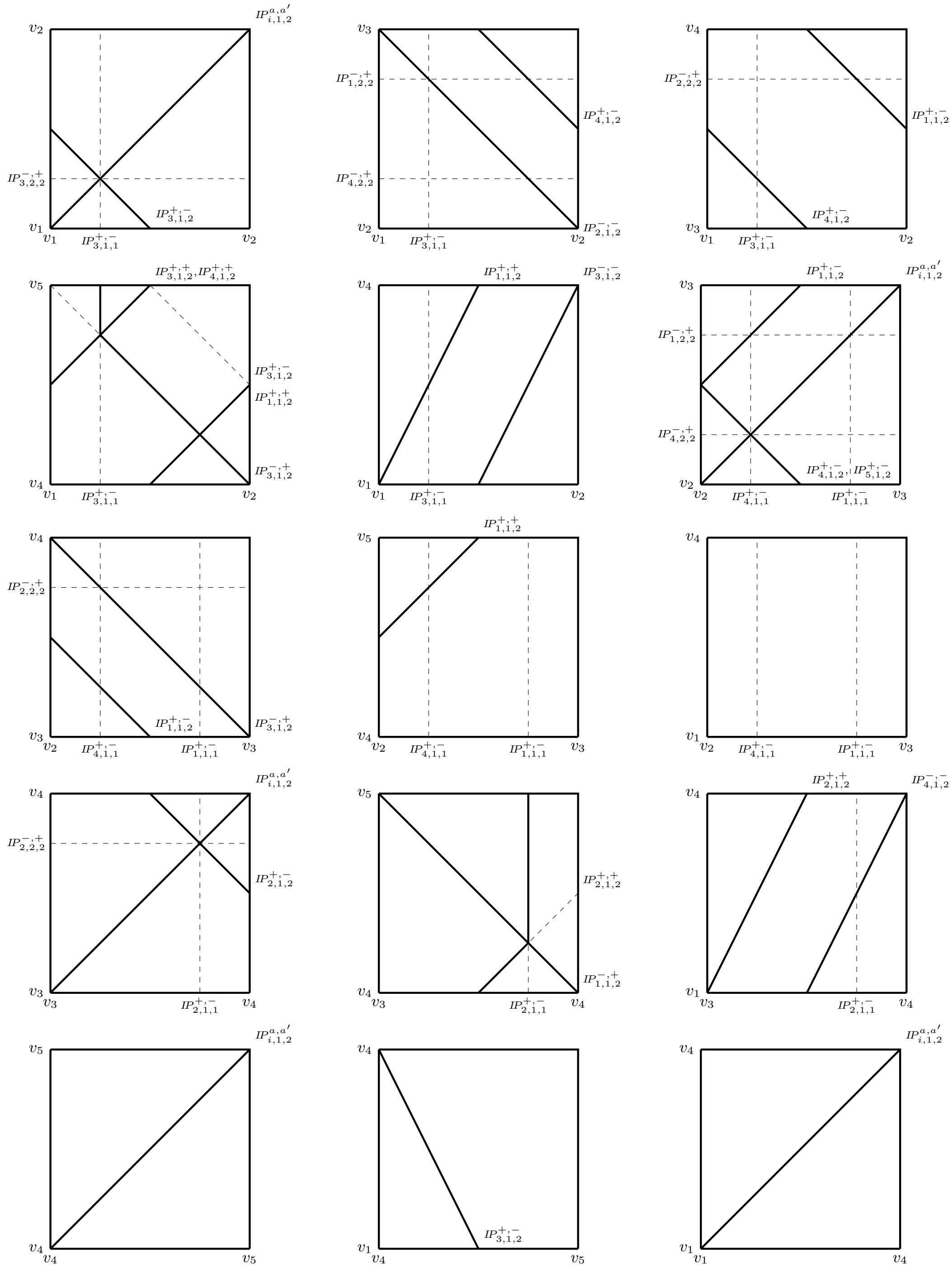


Figure 4.10: Subdivision into linearity domains for all pairs of edges in Example 1. The sets of points where $D_i^a(x_j) = D_i^{a'}(x_{j'})$ for $i \in V$, $a \in \{+,-\}$, $j \in \{1,2\}$ are denoted as $IP_{ijj'}^{aa'}$.

Bibliography

Bitran, G. and Lawrence, K. (1980) “Location service offices: a multicriteria approach”. *Omega*, vol. 8(2): 201–206.

Carrano, E., Takahashi, R., Finseca, C. and Neto, O. (2007) “Bi-objective combined facility location network design”. *EMO 2007, LNCS*, vol. 4403: 486–500.

Colebrook, M. and Sicilia, J. (2007a) “A polynomial algorithm for the multicriteria cent-dian location problem”. *European Journal of Operational Research*, vol. 179(3): 1008–1024.

Colebrook, M. and Sicilia, J. (2007b) “Undesirable facility location problems on multicriteria networks”. *Computers & Operations Research*, vol. 34(5): 1491–1514.

de Berg, M., Cheong, O., van Krefeld, M. and Overmars, M., eds. (2008) *Computational Geometry – Algorithms and Applications.* Springer-Verlag, third edition.

Drezner, Z., ed. (1995) *Facility Location: A Survey of Applications and Methods.* Springer, New York.

Edelsbrunner, H. (1987) *Algorithms in Combinatorial Geometry.* Springer, New York, Berlin, Heidelberg.

Ehrgott, M. (2005) *Multicriteria Optimization.* Springer Berlin Heidelberg.

Ehrgott, M. and Rau, A. (1999) “Bicriteria costs versus service analysis of a distribution network - a case study”. *Journal of Multi-Criteria Decision Analysis*, vol. 8: 256–267.

Eiselt, H. and Laporte, G. (1995) *Facility Loaction: A Survey of Applications and Methods*, chap. Objectives in location problems, pp. 151–180. Springer.

Gale, D. (1963) *Convexity*, vol. VII, chap. Neighborly and cyclic polytopes., pp. 225–232. Proceedings of the Symposia in Pure Mathematics. American Mathematics Society.

Hakimi, S. (1964) “Optimum locations of switching centers and the absolute centers and medians of a graph”. *Operations Research*, vol. 12(3): 450–459.

Hakimi, S. (1965) “Optimum distribution of switching centers in a communications network and some related graph theoretic problems”. *Operations Research*, vol. 13(3): 462–475.

Hamacher, H. W., Labbé, M. and Nickel, S. (1999) “Multicriteria network location problems with sum objectives”. *Networks*, vol. 33(2): 79–92.

Hamacher, H. W., Labbé, M., Nickel, S. and Skriver, A. J. (2002) “Multicriteria semi-obnoxious network location problems (msnlp) with sum and center objectives”. *Annals of Operations Research*, vol. 110: 33–53.

Hershberger, J. (1989) “Finding the upper envelope of n line segments in o(n log n) time”. *Information Processing Letters*, vol. 33(4): 169–174.

Johnson, M. (2001) “A spatial decision support system prototype for housing mobility program planning”. *Journal of Geographical Systems*, vol. 3: 49–67.

Kalcsics, J. (2011) “The multi-facility median problem with pos/neg weights on general graphs”. *Computers & Operations Research*, vol. 38: 674–682.

Kalcsics, J., Nickel, S., Puerto, J. and Rodríguez-Chía, A. M. (2015) “Several 2-facility location problems on networks with equity objectives”. *Networks*, vol. 65(1): 1–9.

Kolokolov, A. and Zaozerskaya, L. (2013) “Solving a bicriteria problem of optimal service centers location”. *Journal of Mathematical Modelling and Algorithms in Operations Research*, vol. 12(2): 105–116.

Nickel, S. and Puerto, J. (2005) *Location Theory: A Unified Approach.* Springer-Verlag, Heidelberg, Germany.

Nickel, S., Puerto, J. and Rodríguez-Chía, A. M. (2005) “MCDM location problems”. In J. A. Figueira, S. Greco and M. Ehrogott, eds., *Multiple Criteria Decision Analysis: State of the Art Surveys*, vol. 78 of *International Series in Operations Research & Management Science*, pp. 761–787. Springer, New York.

Schöbel, A. (2005) “Locating stops along bus or railway lines–a bicriteria problem”. *Annals of Operations Research*, vol. 136(1): 211–227.

Sharir, M. (1994) "Almost tight upper bounds for lower envelopes in higher dimensions". *Discrete & Computational Geometry*, vol. 12: 327–345.

Skriver, A. J., Andersen, K. A. and Holmberg, K. (2004) "Bicriteria network location (bnl) problems with criteria dependent lengths and minisum objectives." *European Journal of Operational Research*, vol. 156: 541–549.

Tansel, B. C., Francis, R. L. and Lowe, T. J. (1982) "Biobjective multifacility minimax location problem on a tree network." *Transportation Science*, vol. 16(4): 407–429.

Zhang, F. and Melachrinoudis, E. (2001) "The maximin-maxisum network location problem". *Computational Optimization and Applications*, vol. 19(2): 209–234.

Chapter 5

Ordered Weighted Average Combinatorial Optimization: Formulations and their properties

ABSTRACT

Multiobjective combinatorial optimization deals with problems considering more than one viewpoint or scenario. The problem of aggregating multiple criteria to obtain a globalizing objective function is of special interest when the number of Pareto solutions becomes considerably large or when a single, meaningful solution is required. Ordered Weighted Average or Ordered Median operators are very useful when preferential information is available and objectives are comparable since they assign importance weights not to specific objectives but to their sorted values. In this chapter, Ordered Weighted Average optimization problems are studied from a modeling point of view. Alternative integer programming formulations for such problems are presented and their respective domains studied and compared. In addition, their associated polyhedra are studied and some families of facets and new families of valid inequalities presented. The proposed formulations are particularized for two well-known combinatorial optimization problems, namely, shortest path and minimum cost perfect matching, and the results of computational experiments presented and analyzed. These results indicate that the new formulations reinforced with appropriate constraints can be effective for efficiently solving medium to large size instances.

Keywords: Combinatorial Optimization, Multiobjective optimization, Weighted Average Optimization, Ordered median.

5.1 Introduction

Multiobjective combinatorial optimization deals with problems considering more than one viewpoint or scenario. They inherit the complexity difficulty of their scalar counterparts, but incorporate additional difficulties derived from dealing with partial orders in the objective function space. The standard solution concept is the set of Pareto solutions. However, the number of Pareto solutions can grow exponentially with the size of the instance and the number of objectives. A first approach to overcome this difficulty focuses on a specific subset of the Pareto set, such as, for instance, the supported Pareto solutions (see, e.g., Ehrgott, 2005). Those are the Pareto solutions that optimize linear scalarizations of the different objectives. However, it is possible to exhibit instances for which even the number of supported solutions grows exponentially with the size of the instance. Furthermore, focusing on supported Pareto solutions a priori excludes compromise solutions that could be preferred by the decision maker. For the above reasons, more involved decision criteria have been proposed in the field of multicriteria decision making (Perny and Spanjaard, 2003). These include objectives focusing on one particular compromise solution, which, for tractability and decision theoretic reasons, seem to be better suited when an appropriate aggregation operator is available.

In some cases, a particularly important Pareto solution related to a weighted ordered average aggregating function is sought. Provided that some imprecise preference information on the objectives is available, and that they are comparable, an averaging operator can be used to aggregate the vector of objective values of feasible solutions. The Ordered Median (OM) objective function is very useful in this context since it assigns importance weights not to specific objectives but to their sorted values. OM operators have been successfully used for addressing various types of combinatorial problems (see, for instance, Ogryczak and Tamir, 2003; Nickel and Puerto, 2005; Puerto and Tamir, 2005; Boland et al., 2006; Marín et al., 2009 or, Fernández et al., 2012).

When applied to values of different objective functions in multiobjective problems, the OM operator is called in the literature Ordered Weighted Average (OWA) (Yager, 1988; Yager and Kacprzyk, 1997). It assigns importance weights to the sorted values of the objective function elements in a multiple objective optimization problem. The OWA has been also used in the literature under the name of Choquet optimization to address continuous problems (Schmeidler, 1986; Lesca et al., 2013) and more recently it has been applied to some combinatorial optimization problems, like the minimum spanning tree and 0-1 knapsack (Galand and Spanjaard, 2012). The OWA is, however, a very broad operator, which, depending on the cases, can be seen as an Ordered Median or as Vector Assignment Ordered Median (Lei and Church, 2012), and which can be applied to any combinatorial optimization problem. We therefore believe that its full potential within combinatorial optimization is worth being exploited. This naturally leads to a thorough study of its modeling properties and alternatives, which is the focus of this chapter.

From a modeling point of view, the OWA operator can be formulated with a combination of discrete and continuous decision variables linked by several families of linear constraints. Since the domain of combinatorial optimization problems can be characterized with ad hoc discrete variables and linear constraints, it becomes clear that every combinatorial optimization problem with an OWA objective can be formulated as a linear integer programming problem, by suitably relating the two sets of variables and constraints. Of course, not all formulations are equally useful. Moreover, it is not even clear that the best formulation for the domain of the combinatorial object should be preferred, because its "integration" with the formulation of the OWA may imply additional difficulties. In this chapter we propose three alternative basic formulations for a combinatorial object with an OWA objective. Each basic formulation uses a different set of decision variables to model the OWA objective. We study properties yielding to alternative formulations, which preserve the set of optimal solutions, and we also compare the formulations among them. In addition we propose various families of facets and valid inequalities, which can be used (independently or in combination) to reinforce the basic formulations. For keeping the length of the chapter within some reasonable limits, we report the results obtained with a particular case of the OWA operator, namely the Hurwicz criterion (Hurwicz, 1951). This criterion, which has been used by other authors in the literature (see, e.g., Ogryczak and Olender, 2012, Galand and Spanjaard, 2012) is a non-monotonic and non-convex criterion. In our experience the Hurwicz criterion behaves quite similarly to other non-convex OWA criteria, so the results we report and derived conclusions can be extended to analogous criteria as well. In the final part of the chapter, we focus on two classical optimization problems: shortest path and minimum cost perfect matching. For these two problems we analyze the empirical performance of the alternative basic formulations and their possible reinforcements and variations. From our computational experience we can not conclude that any of the formulations is superior to the others since the behavior of the proposed formulation varies with the different combinatorial object to be considered (see Section 5.6).

The chapter is structured as follows. Section 5.2 gives the formal definition of the OWA operator and shows that it has as particular cases both the Ordered Median and the Vector Assignment Ordered Median. Section 5.3 presents the three basic formulations, and their variations, for a combinatorial problem with an OWA objective, studies their properties and compares them, whereas Section 5.4 presents different families of valid inequalities and possible reinforcements. Sections 5.5.1 and 5.5.2 respectively present the formulation of the combinatorial object that we use in our empirical study of the shortest path and minimum cost perfect matching problems with an OWA objective. Finally, Section 5.6 describes the computational experiments that we have run and presents and analyzes the obtained numerical results. The chapter ends in Section 5.7 with some conclusions.

5.2 The Ordered Weighted Average Optimization

The Ordered Weighted Average (OWA) operator is defined over a feasible set $Q \subseteq \mathbb{R}^n$. Let $C \in \mathbb{R}^{p\times n}$ be a given matrix, whose rows, denoted by C^i, are associated with the cost vectors of p objective functions. The index set for the rows of C is denoted by $P = \{1, \ldots, p\}$. For $x \in Q$, the vector $y \in \mathbb{R}^p$ is referred to as the outcome vector relative to C. In the following we assume $y = Cx$, with $x \in Q$. For a given y, let σ be a permutation of the indices of $i \in P$ such that $y_{\sigma_1} \geq \ldots \geq y_{\sigma_p}$. Let also $\omega \in \mathbb{R}^{p+}$ denote a vector of non-negative weights. Feasible solutions $x \in Q$ are evaluated with an operator defined as $OWA_{(C,\omega)}(x) = \omega' y_\sigma$. The OWA optimization Problem (OWAP) is to find $x \in Q$ of minimum value with respect to the above operator, that is

$$\text{OWAP:} \quad \min_{x \in Q} OWA_{(C,\omega)}(x)$$

Example 5.1- Consider

$$Q = \left\{x \in \{0,1\}^3 : x_1 + x_2 + x_3 = 2\right\},\ C = \begin{pmatrix} 1 & 4 & 1 \\ \hline 1 & 1 & 3 \\ \hline 5 & 1 & 2 \end{pmatrix} \text{ and } \omega' = \begin{pmatrix} 1 & 2 & 4 \end{pmatrix}.$$

Table 5.1 illustrates, for each feasible $x \in Q$, the values of $y = Cx$, y_σ and $OWA_{(C,\omega)}(x) = \omega' y_\sigma$. The optimal value to the OWAP is $\min_{x \in Q} OWA_{(C,\omega)}(x) = 23$.

x	y	y_σ	$OWA_{(C,\omega)}(x) = \omega' y_\sigma$
$\begin{pmatrix} 1 & 1 & 0 \end{pmatrix}'$	$\begin{pmatrix} 5 & 2 & 6 \end{pmatrix}'$	$\begin{pmatrix} 6 & 5 & 2 \end{pmatrix}'$	24
$\begin{pmatrix} 1 & 0 & 1 \end{pmatrix}'$	$\begin{pmatrix} 2 & 4 & 7 \end{pmatrix}'$	$\begin{pmatrix} 7 & 4 & 2 \end{pmatrix}'$	23
$\begin{pmatrix} 0 & 1 & 1 \end{pmatrix}'$	$\begin{pmatrix} 5 & 4 & 3 \end{pmatrix}'$	$\begin{pmatrix} 5 & 4 & 3 \end{pmatrix}'$	25

Table 5.1: Solutions $x \in Q$, values $y = Cx$, sorted values y_σ and $OWA_{(C,\omega)}(x)$ for Example 5.1.

The OWA operator is a very general function which, as we see below, has as particular cases well-known objective functions. We next describe some of them.

5.2.1 The ordered median objective function (OM).

The OM objective (Nickel and Puerto, 2005) minimizes a weighted sum of ordered elements. It is a well known function that unifies many location problems as the p-median problem, the p-center problem, etc.

Let $Q \subseteq \mathbb{R}^n$ denote the feasible domain for an optimization problem and let $d \in \mathbb{R}^n$ be a cost vector and $\omega \in \mathbb{R}^n$ a given weights vector. For $x \in Q$, let σ denote a permutation of the indices of x, such that $d_{\sigma_j} x_{\sigma_j} \geq d_{\sigma_{j+1}} x_{\sigma_{j+1}}$, $j \in \{1, 2, \ldots, n-1\}$. The OM operator is $OM_{(d,\omega)}(x) = \sum_{j \in P} \omega_j d_{\sigma_j} x_{\sigma_j}$ (note that $p = n$ in this case) and the OM Problem (OMP) is therefore defined as

$$\text{OMP:} \quad \min_{x \in Q} OM_{(d,\omega)}(x) = \sum_{j \in P} \omega_j d_{\sigma_j} x_{\sigma_j}.$$

To cast the OM operator as an OWA operator, we only need to set the rows of the C matrix as $(C^i)' = d_i \mathbf{e^i}$, $i \in \{1, \ldots, n\}$, where $\mathbf{e^i} \in \mathbb{R}^n$ is the i-th unit vector of the canonical basis of $\mathbb{R}^n$. Let $Diag(d)$ denote the diagonal matrix whose diagonal entries are the components of the vector d, thus, $C = Diag(d)$. Then $OM_{(d,\omega)}(x) = OWA_{(Diag(d),\omega)}(x)$.

Example 5.2- Consider

$$Q = \left\{x \in \{0,1\}^3 : x_1 + x_2 + x_3 = 2\right\},\ d = \begin{pmatrix} 5 & 1 & 2 \end{pmatrix}' \text{ and } \omega = \begin{pmatrix} 1 & 2 & 4 \end{pmatrix}'$$

Table 5.2 illustrates, for each feasible $x \in Q$, the values of $(d_j x_j)_{j \in P}$, $(d_{\sigma_j} x_{\sigma_j})_{j \in P}$, and $OM_{(d,\omega)}(x) = \sum_{j \in P} \omega_j d_{\sigma_j} x_{\sigma_j}$. The optimal OM value is $\min_{x \in Q} OM_{(d,\omega)}(x) = 4$.

x	$(d_j x_j)_{j \in P}$	$(d_{\sigma_j} x_{\sigma_j})_{j \in P}$	$OM_{(d,\omega)}(x) = \sum_{j \in P} \omega_j d_{\sigma_j} x_{\sigma_j}$
$\begin{pmatrix} 1 & 1 & 0 \end{pmatrix}'$	$\begin{pmatrix} 5 & 1 & 0 \end{pmatrix}'$	$\begin{pmatrix} 5 & 1 & 0 \end{pmatrix}'$	7
$\begin{pmatrix} 1 & 0 & 1 \end{pmatrix}'$	$\begin{pmatrix} 5 & 0 & 2 \end{pmatrix}'$	$\begin{pmatrix} 5 & 2 & 0 \end{pmatrix}'$	9
$\begin{pmatrix} 0 & 1 & 1 \end{pmatrix}'$	$\begin{pmatrix} 0 & 1 & 2 \end{pmatrix}'$	$\begin{pmatrix} 2 & 1 & 0 \end{pmatrix}'$	4

Table 5.2: Solutions $x \in Q$, values $d_j x_j$, sorted $d_{\sigma_j} x_{\sigma_j}$ and $OM_{(d,\omega)}(x)$ for the OM of Example 5.2.

To cast the OM operator as an OWA operator, we only need to set the rows of the C matrix as

$$C = Diag(d) = \begin{pmatrix} 5 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 2 \end{pmatrix}.$$

The values of $y = Cx$, y_σ and $OWA_{(C,\omega)}(x) = \omega' y_\sigma$ are shown in Table 5.3. The optimal OWA value is $\min_{x \in Q} OWA_{(Diag(d),\omega)}(x) = \min_{x \in Q} OM_{(d,\omega)}(x) = 4$.

x	y	y_σ	$OWA_{(C,\omega)}(x) = \omega' y_\sigma$
$\begin{pmatrix} 1 & 1 & 0 \end{pmatrix}'$	$\begin{pmatrix} 5 & 1 & 0 \end{pmatrix}'$	$\begin{pmatrix} 5 & 1 & 0 \end{pmatrix}'$	7
$\begin{pmatrix} 1 & 0 & 1 \end{pmatrix}'$	$\begin{pmatrix} 5 & 0 & 2 \end{pmatrix}'$	$\begin{pmatrix} 5 & 2 & 0 \end{pmatrix}'$	9
$\begin{pmatrix} 0 & 1 & 1 \end{pmatrix}'$	$\begin{pmatrix} 0 & 1 & 2 \end{pmatrix}'$	$\begin{pmatrix} 2 & 1 & 0 \end{pmatrix}'$	4

Table 5.3: The OM instance of Example 5.2 as an OWA: $y = Cx$, y_σ and $OWA_{(C,\omega)}(x)$.

5.2.2 The vector assignment ordered median objective function.

The Vector Assignment Ordered Median (VAOM) problem was recently introduced by Lei and Church (2012) in the context of discrete location-allocation problems. In this context, the VAOM generalizes both OM and Vector Assignment Median (Weaver and Church, 1985). As we see below the OWA generalizes the VAOM as well. First, we briefly introduce the VAOM.

The main decisions in location-allocation problems are the set of facilities to open, and the assignment of customers to opened facilities so as to satisfy their demand. Consider a given set of customers $P = \{1, \dots, p\}$, where each customer is also a potential location for a facility, and let $q \leq p$ denote the number of facilities to open. Associated with each customer $i \in P$ there is a demand a_i. A unit of demand at customer i served from facility k incurs a cost d_{ik}. We will use $\mathbf{d}^i$ to denote the p dimensional vector of the distances associated with customer i. Usual objectives focus on service cost minimization.

Many location-allocation models allow splitting the demand at customers among several facilities, so allocating customer i to facility k means that some positive fraction of a_i is served from facility k. However, without any further incentive or constraint, in optimal solutions customers will be allocated to one single facility, the closest one among those that are open. Since such solutions often exhibit privileged customers, equity measures have been proposed to balance out the service level of the customers. This is the case of the VAOM that imposes the specific fractions of the demand at each customer to be served from the various open facilities. Let $\gamma_{i\ell}$ denote the fraction of a_i that must be served from the ℓ-th closest facility to customer i where $\ell \in I = \{1, ..., q\}$. To measure the service level of customer i in a given solution, the distances from i to the different open facilities are ordered and weighted with the values $\gamma_{i\ell}$ according to their rank in the sorted list of distances. This invites to characterize solutions by means of binary decision variables $x^i_{k\ell}$, $i, k \in P$, $\ell \in I$, where $x^i_{k\ell}$ is equal to 1 if i is allocated to facility k as the ℓ-th closest facility. Now, the service cost of customer i can be computed as $s_i = \sum_{k\in P}\sum_{\ell\in I} a_i\gamma_{i\ell}d_{ik}x^i_{k\ell}$. Note that s_i can be expressed in a compact way as $s_i = \overline{C}^i\mathbf{x}^i$, where $\mathbf{x}^i$ is the vector of decision variables $(x^i_{k\ell})_{k\in P,\ell\in I} = (x^i_{11}, x^i_{12}, ..., x^i_{21}, x^i_{22}, ...)'$, and $(\overline{C}^i)' = (a_i\gamma_{il}d_{ik})_{k\in P,\ell\in I}$.

The VAOM operator is computed as a weighted sum of the service costs of all customers. A weight ω_j is applied to the customer with the j-th lowest service level, i.e. with the j-th highest service cost. For a given solution, x, and its associated vector s as defined above, let σ be a permutation of the indices of P such that $s_{\sigma_1} \geq \ldots \geq s_{\sigma_p}$. Then, $VAOM_{(d,\gamma,a,\omega)}(x) = \sum_{j=1}^{p}\omega_j s_{\sigma_j}$ and the VAOM Problem (VAOMP) is therefore defined as

$$\text{VAOMP:} \quad \min_{x\in Q} VAOM_{(d,\gamma,a,\omega)}(x) = \sum_{j=1}^{p}\omega_j s_{\sigma_j}.$$

The set of feasible solutions to the problem is fully characterized by the set of feasible assignments, since an explicit representation of the open facilities is not needed. These can be obtained directly from x by identifying the indices $k \in P$ with $x^i_{k\ell} = 1$ for some $i \in P$, $\ell \in I$. Thus in this problem Q is given by the set of feasible assignments. For reasons that will become evident when we cast the VAOM operator as an OWA, we express the assignment vectors x as one dimensional n vectors with $n = p^2q$. In particular x is partitioned in p blocks, each of them associated with a different customer $i \in P$. That is, $x = (\mathbf{x}^{1'}| \ldots | \mathbf{x}^{i'} | \ldots |\mathbf{x}^{p'})'$. In turn, each block $\mathbf{x}^i$ consists of p smaller blocks, each with q components. The k-th block of $\mathbf{x}^i$ contains the q components $x^i_{k\ell}$ for the indices $\ell \in I$.

Now, to cast the VAOM as an OWA operator, we define p objective functions $\overline{C}^i\mathbf{x}^i$, one associated with each customer $i \in P$. In particular, objective $\overline{C}^i\mathbf{x}^i$ represents the service cost of customer $i \in P$, s^i. With the above characterization of vectors $x \in Q$, each $\overline{C}^i$ must be defined by a n vector. Thus expressing the VAOM as an OWA becomes basically a notation issue. For each fixed $i \in P$, again we partition the cost vector $\overline{C}^i$ in p blocks. Similarly to the partition of vectors $x \in Q$, each block corresponds to a different customer, and has pq components. We now set at value 0 all the entries except those in the block of customer i, which are given by the entries of the vector $\overline{C}^i$ as defined above. That is: $C^i = (\mathbf{0}_{pq} \mid \ldots \mid \overline{C}^i \mid \ldots \mid \mathbf{0}_{pq})$, where $\mathbf{0_{pq}} = (0, ..., 0) \in \mathbb{R}^{pq}$. With this notation it becomes clear that $C^ix = \overline{C}^i\mathbf{x}^i$. Hence,

$$VAOM_{(d,\gamma,a,\omega)}(x) = OWA_{(C,\omega)}(x).$$

Example 5.3- Consider an instance of a VAOM problem with $p = 3$ customers in which $q = 2$ facilities must be opened. Suppose all the customers have one unit of demand, i.e. $a_1 = a_2 = a_3 = 1$, and suppose the rest of the data is the following:

$$(d_{ik})_{i,k\in P} = \begin{pmatrix} 0 & 2 & 6 \\ 2 & 0 & 4 \\ 8 & 4 & 0 \end{pmatrix}, \ (\gamma_{il})_{i\in P, l\in I} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \\ 1 & 0 \end{pmatrix}, \ \omega' = \begin{pmatrix} 0 & 1 & 2 \end{pmatrix}.$$

Since $q = 2$ the feasible combinations of facilities to open are $\{1,2\}$, $\{1,3\}$ and $\{2,3\}$. When the distances of each customer to the potential facilities are all different, like in this example, each combination of open facilities determines a unique feasible assignment vector x. For instance, when facilities 1 and 2 open, then customer 1, has facility 1 as the closest and facility 2 as the second closest, so $x^1_{11} = x^1_{22} = 1$, and $x^1_{12} = x^1_{21} = x^1_{31} = x^1_{32} = 0$. The service cost of customer 1 is thus $s_1 = \gamma_{11}d_{11}x^1_{11} + \gamma_{12}d_{12}x^1_{22} = 0 + 0.5 \times 2 = 1$. For customer 2 we have $x^2_{12} = x^2_{21} = 1$, and $x^2_{11} = x^2_{22} = x^2_{31} = x^2_{32} = 0$, with service cost $s_2 = 0 + 0.5 \times 2 = 1$. With this set of open facilities, the assignment for customer 3 is $x^3_{12} = x^3_{21} = 1$, and $x^3_{11} = x^3_{22} = x^3_{31} = x^3_{32} = 0$ with service cost $s_3 = 4$. Since $s_3 \geq s_1 \geq s_2$ the objective function value for this solution is thus $0\times 4 + 1\times 1 + 2\times 1 = 3$.

Proceeding similarly with the other possible combinations of open facilities we obtain the complete set of feasible solutions Q, which in this example is given by the set of binary vectors given in Table 5.4:

x^1_{11}	x^1_{12}	x^1_{21}	x^1_{22}	x^1_{31}	x^1_{32}	x^2_{11}	x^2_{12}	x^2_{21}	x^2_{22}	x^2_{31}	x^2_{32}	x^3_{11}	x^3_{12}	x^3_{21}	x^3_{22}	x^3_{31}	x^3_{32}
1	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0
1	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	1	0

Table 5.4: Complete set of feasible solutions Q as binary vectors for Example 5.3.

For modeling the VAOM as an OWA we define the cost matrix C as:

$$C = \left(\begin{array}{cc|cc|cc||cc|cc|cc||cc|cc|cc} 0 & 0 & 1 & 1 & 3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 4 & 0 & 0 & 0 \end{array}\right).$$

Table 5.5 shows the values of y, y_σ and $OWA_{(C,\omega)}(x)$ for each $x \in Q$. The optimal value of the VAOM is $\min_{x\in Q} VAOM_{(d,\gamma,a,\omega)}(x) = \min\{3,3,2\} = 2$.

y	y_σ	$OWA_{(C,\omega)}(x) = \omega' y_\sigma$
$\begin{pmatrix} 1 & 1 & 4 \end{pmatrix}'$	$\begin{pmatrix} 4 & 1 & 1 \end{pmatrix}'$	3
$\begin{pmatrix} 3 & 3 & 0 \end{pmatrix}'$	$\begin{pmatrix} 3 & 3 & 0 \end{pmatrix}'$	3
$\begin{pmatrix} 4 & 2 & 0 \end{pmatrix}'$	$\begin{pmatrix} 4 & 2 & 0 \end{pmatrix}'$	2

Table 5.5: Values of y, y_σ and $OWA_{(C,\omega)}(x)$ for the feasible solutions of Example 5.3.

5.2.3 The Vector Assignment Ordered Median function of an abstract combinatorial optimization problem

In the previous section we have applied the VAOM operator to the locations and allocations of a general multifacility location problem, according to the original definition by Lei and Church (2012). Nevertheless, this operator can be also applied to the characteristic vector of a combinatorial solution of any abstract combinatorial optimization problem, as we also did with the ordered median operator. In doing that we obtain a more general interpretation of this type of objective function that can also be cast within the OWA operator.

Let $Q \subseteq \mathbb{R}^n$ denote the feasible domain for an optimization problem, $\omega \in \mathbb{R}^{p+}$ a given vector of nonnegative weights and $P = \{1, \dots, p\}$. Recall that a VAOM operator considers for each objective function s_i, $i \in P$ different fractions, γ^i, of the cost vector d for the sorted elements of the decision vector x.

For $x \in Q$, the evaluation of the i-th component of the VAOM objective is given by $s_i = \gamma^i d_i x_i$, for all $i \in P$. Let σ denote a permutation of the indices of P, such that $s_{\sigma_i} \geq s_{\sigma_{i+1}}$, for $i = 1, \dots, p-1$. The VAOM operator is $VAOM_{(d,\gamma,\omega)}(x) = \sum\limits_{i \in P} \omega_i s_{\sigma_i}$. The reader may note that the original definition of VAOM can be accommodated to this general setting once we identify the combinatorial object Q as the set of location-allocations in the discrete location problem. In that case, there are $i = 1, \dots, n$ objective functions associated with each of the customers and then the fractions that apply to each customer i are non-null only for a subset of the open facilities (servers) corresponding to the q-closest ones.

This can be done by defining a set of variables, one per customer i, with n blocks. In the block k, $x^i_{\bullet k} = (x^i_{1k}, ..., x^i_{nk})'$ accounts for the allocation of i to any facility as the k-th closest, therefore $x^i = (x^{i\prime}_{\bullet 1} \mid x^{i\prime}_{\bullet 2} \mid ... \mid x^{i\prime}_{\bullet n})'$ for $i = 1, \dots, n$. This way, the cost vectors must also have the same structure by blocks, each block corresponding with the level of assignment, i.e. denoting by $d^i_{\bullet} = (d^i_1, ..., d^i_n)' \in \mathbb{R}^{n^2}$ then $d^i = (d^{i\prime}_{\bullet} \mid d^{i\prime}_{\bullet} \mid ... \mid d^{i\prime}_{\bullet})'$. Finally, since the fractions of costs are applied according to the level of assignment, the structure of the vector of fractions γ^i is also by blocks. Block k represents the fraction of the cost that is accounted for costumer i at the k-th level of assignment. Denoting by $\gamma^i_\ell = (\gamma_{i\ell}, ..., \gamma_{i\ell})' \in \mathbb{R}^n$ then $\gamma^i = (\gamma^i_1 \mid \gamma^{i\prime}_2 \mid ... \mid \gamma^{i\prime}_n)'$ for $i = 1, \dots, n$.

To cast the VAOM as an OWA operator, we only need to set $\bar{\gamma}^i = (\underbrace{\mathbf{0}_{np}}_{1} \mid \dots \mid \underbrace{\gamma^{i\prime}}_{i} \mid \dots \mid \underbrace{\mathbf{0}_{np}}_{p})'$, $\bar{d^i} = (\underbrace{\mathbf{0}_{np}}_{1} \mid \dots \mid \underbrace{d^{i\prime}}_{i} \mid \dots \mid \underbrace{\mathbf{0}_{np}}_{p})'$, $x = (\mathbf{x}^{1\prime} | \dots | \mathbf{x}^{i\prime} | \dots | \mathbf{x}^{p\prime})'$ and $C^i = (\bar{\gamma}^i_j \bar{d}^i_j)_{j=1}^{p(pn)}$. Then, the

VAOM can be written as the following OWA operator $VAOM_{(d,\gamma,\omega)}(x) = OWA_{(C,\omega)}(x)$.

As we have shown above, OWA is a very general operator. In the following, we will work in more particular settings, namely we shall restrict ourselves to assume that Q is a combinatorial object which can be represented by a system of linear inequalities.

5.3 Basic formulations for the OWAP, properties and reinforcements

This section presents alternative Mixed Integer Programming (MIP) formulations for an OWAP, which are analyzed and compared. The starting point of our study are three basic formulations, which, broadly speaking, differ from one to another on how the permutation that defines the ordering of the cost function values is modeled. Two of the formulations presented use binary variables z to define the *specific positions* in the ordering of the sorted cost function values, whereas the other one uses binary variables s to define the *relative position* in the ordering of the sorted cost function values. One of the two formulations based on the z variables also uses an additional set of decision variables y for modeling the specific values of the cost functions depending on their position in the ordering. All three formulations use a set of decision variables θ to compute the values of the objectives sorted at specific positions. In each case, alternative formulations are presented, which preserve the set of optimal solutions. Before addressing any concrete formulation we discuss the meaning of both sets of variables z and s as well as their relationships.

5.3.1 Alternative formulations for permutations

The essential element in our formulations rests on the representation of ordering within a MIP model. To such end, we devote this section to describe how a permutation can be represented with binary variables. Recall that we have introduced $P = \{1, \dots, p\}$ as the set of the cost function indices. Let $\pi : P \to P$ be a function representing a permutation of P. That is, it assigns the index i of each cost function (also denoted by *cost function* i) to a position indexed by j (also denoted by *position* j). Note that π is a permutation if each cost function is assigned to a single position and if each position contains a single cost function index. In what follows, we use $\pi_i = \pi(i)$ to denote the position occupied by cost function $i \in P$ and $\sigma_j = \pi^{-1}(j)$ to denote the index of the cost function that occupies position j (we recall that the notation σ was previously used in Section 5.2). Note that σ also defines a permutation of the positions of P. In what follows we will indistinctively use π and σ. Slightly abusing notation we refer to π as to *the cost functions permutation* and to its inverse σ as to the *positions permutation*.

In order to model π as a permutation, let z_{ij} be a binary decision variable defined as

$$z_{ij} = \begin{cases} 1 & \text{if cost function } i \text{ occupies position } j, \text{ (i.e. if } \pi_i = j) \\ 0 & \text{otherwise.} \end{cases}$$

The set of variables z defines a permutation if:

(i) each position contains a single cost function:

$$\sum_{i \in P} z_{ij} = 1 \qquad j \in P, \tag{5.1}$$

and,

(ii) each cost function i is assigned to a single position j:

$$\sum_{j \in P} z_{ij} = 1 \qquad i \in P. \tag{5.2}$$

In addition, we observe that since system (5.1)–(5.2) contains exactly $2p-1$ linearly independent equations, the above permutation can also be represented without variables z_{i1}, for all $i \in P$, that can be replaced by $1 - \sum_{j \in P: j>1} z_{ij}$. In this way, system (5.1)–(5.2) can also be rewritten as

$$\sum_{i \in P} z_{ij} = 1 \qquad j \in P : j > 1, \tag{5.3}$$

$$\sum_{j \in P: j>1} z_{ij} \leq 1 \qquad i \in P. \tag{5.4}$$

Example 5.4- Let π be a permutation defined by $\pi = \begin{pmatrix} 3 & 2 & 4 & 1 \end{pmatrix}$ or equivalently by $\sigma = \begin{pmatrix} 4 & 2 & 1 & 3 \end{pmatrix}$. Then, π can be represented by using variables z as follows:

$$(z_{ij})_{i,j \in P} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \text{ or } (z_{ij})_{i,j \in P: j>1} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

□

An alternative representation of a permutation, which we have also found useful is based on a different set of variables defined as:

$$s_{ij} = \begin{cases} 1 & \text{if cost function } i \text{ is placed before position } j \text{ in the ordering,} \\ 0 & \text{otherwise.} \end{cases}$$

The set of variables s defines a permutation if:

(*i*) for all $j \in P$ there are $j-1$ cost functions placed before position j:

$$\sum_{i \in P} s_{ij} = j - 1 \qquad j \in P, \tag{5.5}$$

and

(*ii*) cost function i cannot be placed before position j unless it is also placed before position $j+1$, i.e.,

$$s_{ij+1} - s_{ij} \geq 0 \qquad i, j \in P : j < p. \tag{5.6}$$

Again we can reduce the number of decision variables, now by eliminating s_{i1} for all $i \in P$. Indeed, since there is no cost function placed before position 1 in any ordering, all the $s_{i1}, i \in P$ can be fixed to zero. In this way, permutation (5.5)–(5.6) can be also represented by means of the following reduced set of constraints:

$$\sum_{i \in P} s_{ij} = j - 1 \qquad j \in P : j > 1, \tag{5.7}$$

$$s_{ij+1} - s_{ij} \geq 0 \qquad i, j \in P : 1 < j < p. \tag{5.8}$$

Example 5.5- Let π be a permutation defined by $\pi = \begin{pmatrix} 3 & 2 & 4 & 1 \end{pmatrix}$ or equivalently by $\sigma = \begin{pmatrix} 4 & 2 & 1 & 3 \end{pmatrix}$. Then, π can be represented by using variables s as follows:

$$(s_{ij})_{i,j \in P} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \text{ or } (s_{ij})_{i,j \in P : j > 1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

□

With the above considerations, variables z and s are related by means of

$$z_{ij} = \begin{cases} s_{ij+1} - s_{ij} & i \in P, j = 1, ..., p-1 \\ 1 - s_{ij} & i \in P, j = p \end{cases} \tag{5.9}$$

and equivalently,

$$s_{ij} = 1 - \sum_{k \geq j} z_{ik},\ i, j \in P. \tag{5.10}$$

5.3.2 *OWAP* formulations with variables for the positions of sorted cost function values

For a given feasible set $Q \subseteq \mathbb{R}^n$, consider the binary decision variables z as defined in Section 5.3.1 to represent the permutation π associated with the sorted cost function values $C^i x$, $i \in P$. Let also θ_j be a real decision variable equal to the value of the cost function sorted in position j. Next, we give an integer linear programming description of the *OWAP* where we use M to denote a non-negative upper bound of the value of all the cost functions. (We refer the interested reader to Boland et al. (2006) or Nickel and Puerto (2005) for similar sets of decision variables and formulations for the discrete ordered median location problem.)

$$F_0^z: \quad V = \min \sum_{j \in P} \omega_j \theta_j \tag{5.11a}$$

$$s.t. \quad \sum_{i \in P} z_{ij} = 1 \qquad j \in P \tag{5.11b}$$

$$\sum_{j \in P} z_{ij} = 1 \qquad i \in P \tag{5.11c}$$

$$C^i x \leq \theta_j + M(1 - z_{ij}) \qquad i, j \in P \tag{5.11d^0}$$

$$\theta_j \geq \theta_{j+1} \qquad j \in P : j < p \tag{5.11e}$$

$$x \in Q, z \in \{0,1\}^{p \times p} \tag{5.11f}$$

The objective function (5.11a) minimizes the weighted average of sorted objective function values, provided that θ_j, $j \in P$, are enforced to take on the appropriate values. As seen, constraints (5.11b)-(5.11c) define a cost functions permutation by placing at each position of π a single cost function and each cost function at a single position of π. Constraints (5.11d^0) relate cost function values with the values placed in a sorted sequence. Constraint (5.11e) imposes that the sorted values are ordered non-increasingly.

In the following we denote by Ω_0^z the domain of feasible solutions to formulation F_0^z. That is,

$$\Omega_0^z = \left\{(x, z, \theta) \text{ satisfying constraints (5.11b), (5.11c), (5.11d}^0\text{), (5.11e), (5.11f)}\right\}.$$

Consider now the family of inequalities

$$C^i x \leq \theta_j + M(1 - \sum_{k \geq j} z_{ik}) \qquad i, j \in P, \tag{11d}$$

and note that, for z satisfying (5.11c), inequalities (11d) can be rewritten as

$$C^i x \leq \theta_j + M \sum_{k<j} z_{ik} \qquad i, j \in P, \tag{11d'}$$

since for all $i, j \in P$, $1 - \sum_{k \geq j} z_{ik} = \sum_{k<j} z_{ik}$.

Remark 5.1- Observe that when variables z_{i1}, $i \in P$ are not defined and the permutation is described by means of inequalities (5.3) and (5.4), then constraints ($5.11d^0$), (11d) and (11d') must consider separately the case $j = 1$ from the case $j \in P, j > 1$. In particular, the case $j = 1$ reduces to

$$C^i x \leq \theta_1 \qquad i \in P, \tag{5.12}$$

since the first position has always a value greater than or equal to any cost function.

Let $\Omega^z = \{(x, z, \theta)$ satisfying constraints (5.11b), (5.11c), (11d), (5.11e), (5.11f)$\}$ denote the domain obtained from Ω_0^z when constraints ($5.11d^0$) are replaced by constraints (11d).

Property 5.1- $\Omega_0^z = \Omega^z$.

`Proof.`
It is clear that $\Omega_0^z \supseteq \Omega^z$, since for $i, j \in P$ given, the right hand side of the associated constraint (11d) is smaller than or equal to that of constraint ($5.11d^0$).

To prove that $\Omega_0^z \subseteq \Omega^z$ also holds let $(x, z, \theta) \in \Omega_0^z$ and we show that (x, z, θ) satisfies constraints (11d). For $i, j \in P$ given, we distinguish two cases:

- If $z_{ij} = 1$ then (11d) holds for this pair of indices.
- If $z_{ij} = 0$ then by (5.11c), there must exist $j' \in P$, $j' \neq j$, such that $z_{ij'} = 1$. If $j' < j$, then $\sum_{k \geq j} z_{ik} = z_{ij} = 0$, and (11d) holds for the pair of indices i, j. Otherwise, if $j' > j$, then $\sum_{k \geq j} z_{ik} = z_{ij'} = 1$ so the right hand side of constraint (11d) for the pair i, j takes the value θ_j. Now constraint ($5.11d^0$) for the pair of indices i, j' implies that $C^i x \leq \theta_{j'}$. By constraints (5.11e), we also have $\theta_j \geq \theta_{j'}$ and thus (11d) also holds for the pair of indices i, j. □

Remark 5.2- Since $\Omega_0^z = \Omega^z$, an equivalent formulation for the OWAP is

$$F^z: \quad V = \min \sum_{j \in P} \omega_j \theta_j$$
$$s.t. \ (x, z, \theta) \in \Omega^z.$$

Formulation F^z can be preferred to formulation F_0^z for solving an OWAP, since it may provide tighter linear programming bounds, given that, for fractional vectors z satisfying constraints (5.11b)-(5.11c), constraints (5.11d^0) are dominated by constraints (11d).

In the search for optimal solutions to the OWAP any formulation whose optimal solution set coincides with that of the OWAP can be of interest. Such formulations could be preferred because they use fewer variables or constraints, or because their feasible domain has a structure which is easier to explore. Next we present three such formulations. All of them can be seen as relaxations of formulation F^z in the sense that their feasible domains contain Ω^z. However, all of them are valid formulations for the OWAP since they preserve the set of optimal solutions of F^z, i.e. their set of optimal solutions coincides with that of F^z. First we prove a property of optimal solutions.

Lemma 5.1- Let $(x^*, z^*, \theta^*) \in \Omega^z$ be an optimal solution to F^z. Then for each $j \in P$ there exists $i \in P$ with $\theta_j^* = C^i x^*$.

`Proof.`
Let $\tilde{x}$ be a feasible solution in Q. Then, there exists a positions permutation σ that sorts the cost functions values in non-increasing order. That is, $C^{\sigma_j}\tilde{x} \geq C^{\sigma_{j+1}}\tilde{x}, \forall j \in P \setminus \{p\}$. Therefore, we can set $\tilde{z} = (z_{\sigma_j,j})_{j \in P}$ and $\theta = (C^{\sigma_j}\tilde{x})_{j \in P}$. Since this is true for each $x \in Q$, it is true in particular for x^*. □

From the above lemma, we observe that Ω^z is always non empty, provided that Q is non empty.

Let $\Omega_{R1}^z = \{(x, z, \theta)$ satisfying constraints (5.11b), (5.11c), (11d), (5.11f)$\}$, i.e, Ω_{R1}^z is the relaxation of the domain Ω^z once the set of constraints (5.11e) is removed. Next, consider the formulation

$$F_{R1}^z: \quad V = \min \sum_{j \in P} \omega_j \theta_j$$
$$s.t. \ (x, z, \theta) \in \Omega_{R1}^z.$$

Property 5.2- Every optimal solution to F_{R1}^z is also optimal to F^z.

`Proof.`
Since $\Omega^z \subseteq \Omega_{R1}^z$it is enough to prove that every optimal solution to F_{R1}^z is feasible to F^z. Let $(x, z, \theta) \in \Omega_{R1}^z$ be an optimal solution to F_{R1}^z and σ a permutation that sorts the cost function values of x. Let us see that θ verifies constraint (5.11e).

If $i \geq j$ then $C^{\sigma_i}x$ occupies, in the sorted sequence of objective values, a position greater than or equal to the j-th. Thus, Constraint (11d) implies that $\theta_j \geq \max_{i \geq j} C^{\sigma_i}x$. Since we are minimizing a function which is a linear combination with non-negative weights of the θ variables, it follows that in any optimal solution $\theta_j = \max_{i \geq j} C^{\sigma_i}x$ since, otherwise, the value of θ_j could be decreased to $\max_{i \geq j} C^{\sigma_i}x$, while keeping all other variables values unchanged and without increasing the objective function value. Therefore (5.11e) holds since, otherwise, there would exist j' such that $\theta_{j'+1} > \theta_{j'} \Leftrightarrow \max_{i \geq j'+1} C^{\sigma_i}x > \max_{i \geq j'} C^{\sigma_i}x$ which is not possible. □

Consider now $\Omega^z_{R2} = \{(x, z, \theta)$ satisfying constraints (5.11b), (11d), (5.11f)$\}$, i.e, Ω^z_{R2} is the relaxation of the domain Ω^z_{R1} once the set of constraints (5.11c) is removed. Next, consider the formulation

$$F^z_{R2}: \quad V = \min \sum_{j \in P} \omega_j \theta_j$$
$$s.t. \;\; (x, z, \theta) \in \Omega^z_{R2}.$$

Property 5.3- Every optimal solution to F^z_{R2} is also optimal to F^z.

Proof.

Since $\Omega^z \subseteq \Omega^z_{R2}$ it is enough to prove that any optimal solution to F^z_{R2} is feasible to F^z. Let (x, z, θ) be an optimal solution to F^z_{R2}. If (x, z, θ) is optimal to F^z_{R1} then, by using Property 5.2, (x, z, θ) is also optimal to F^z. Thus, to prove that (x, z, θ) is optimal to F^z, it suffices to prove that (x, z, θ) satisfies inequalities (5.11c).

We prove first that $\sum_{j \in P} z_{ij} \leq 1$ for all $i \in P$. Using the notation $r_{ij} = \sum_{k \geq j} z_{ik}$, for all $i, j \in P$, constraints (11d) can be rewritten as

$$C^i x \leq \theta_j + M(1 - r_{ij}) \Leftrightarrow \theta_j \geq C^i x + M(r_{ij} - 1).$$

Therefore, for all $j \in P$,

$$\theta_j = \max_{i \in P}\{C^i x + M(r_{ij} - 1)\}.$$

Suppose there exists $i' \in P$ with $\sum_{j \in P} z_{i'j} = r > 1$, and let $j' = \arg\max\{r_{i'j} = 2 \mid j \in P\}$. If several indices exist with $\sum_{j \in P} z_{ij} > 1$ we select i' as the one with maximum associated j'.

The criterion for the selection of i' and the definition of j' imply that $r_{i'j'} = 2$ and $r_{ij'} \leq 1$ for all $i \neq i'$.

Therefore, since M is a strict upper bound on the value of any cost function, the actual value of $\theta_{j'}$ is determined by cost function i', and we have

$$\theta_{j'} = C^{i'}x + M(r_{i'j'} - 1) = C^{i'}x + M.$$

Also, $r_{i'j} \geq 2$ for all $j < j'$. Thus, $\theta_j \geq C^{i'}x + M$ for all $j < j'$. Furthermore, $r_{ij} \leq 1$ for all $i \in P$, $j > j'$, implying that $\theta_j < M$ for all $j > j'$.

Observe, on the other hand, that $\sum_{j\in P} z_{i'j} > 1$ implies that there exists some $i'' \in P$, $i'' \neq i'$ with $\sum_{j\in P} z_{i''j} = 0$. (Otherwise, adding up all constraints (5.11b) we get a contradiction.)

Let us now define the solution $(x, \overline{z}, \overline{\theta}) \in \Omega^z_{R2}$ with the same x components as above, where

$$\overline{z}_{ij} = \begin{cases} 0 & \text{if } i = i', \text{ and } j = j' \\ 1 & \text{if } i = i'', \text{ and } j = j' \\ z_{ij} & \text{otherwise.} \end{cases}$$

It is clear that $\sum_{j\in P} \overline{z}_{i'j} = r - 1$, and, $\sum_{k\geq j} \overline{z}_{i'k} = r_{i'j} - 1$, for all $j \leq j'$. It is also clear that $\sum_{j\in P} \overline{z}_{i''j} = 1$, and, $\sum_{k\geq j} \overline{z}_{i''k} = 1$, for all $j \leq j'$, and 0 for $j > j'$. For all other $i \neq i', i''$, it holds that $\sum_{j\in P} \overline{z}_{i'j} = \sum_{j\in P} z_{i'j}$. Since $\sum_{k\geq j'} \overline{z}_{ik} \leq 1$, for all $i \in P$ we now have

$$\overline{\theta}_{j'} = \max_{i\in P}\{C^i x + M(\sum_{k\geq j'} \overline{z}_{ik} - 1)\} < M \leq C^{i'}x + M = \theta_{j'},$$

and, $\overline{\theta}_j \leq \theta_j$, for all $j \neq j'$.

Therefore, since we are minimizing a linear function with non-negative weights of the θ variables, the objective function value of $(x, \overline{z}, \overline{\theta})$ is smaller than that of (x, z, θ), contradicting the optimality of (x, z, θ). Hence, $\sum_{j\in P} z_{ij} \leq 1$ for all $i \in P$.

Let us, finally, see that $\sum_{j\in P} z_{ij} \neq 0$ for all $i \in P$. Assume on the contrary that $\sum_{j\in P} z_{i'j} = 0$ for some $i' \in P$. Then, by adding up all constraints (5.11b) we get $p = \sum_{j\in P} \left(\sum_{i\in P} z_{ij}\right) = \sum_{i\in P} \left(\sum_{j\in P} z_{ij}\right) = \sum_{i\in P, i\neq i'} \left(\sum_{j\in P} z_{ij}\right) \leq p - 1$, which is impossible. □

We now consider the inequality version of constraints (5.11b)

$$\sum_{i\in P} z_{ij} \leq 1 \qquad j \in P. \tag{5.11b$_\leq$}$$

Remark 5.3- Observe that if we replace inequalities (5.11b) by (5.11b$_\leq$), constraints (11d) are no longer equivalent to (11d').

Let us define the domain $\Omega^z_{R3} = \{(x, z, \theta)$ satisfying constraints (5.11b$_\leq$), (11d'), (5.11f)$\}$.

It is clear that $\Omega^z \subseteq \Omega^z_{R3}$. However, as we next see, both sets are equivalent for the minimization of the objective (5.11a) in the sense that they define the same set of optimal solutions. Consider the problem

$$F_{R3}^z \quad V = \min \sum_{j \in P} \omega_j \theta_j$$

$$s.t. \ \ (x, z, \theta) \in \Omega_{R3}^z.$$

Lemma 5.2- $\Omega_{R2}^z \subseteq \Omega_{R3}^z$.

Proof.

We prove that any feasible solution $(x, z, \theta) \in \Omega_{R2}^z$ verifies that $(x, z, \theta) \in \Omega_{R3}^z$. To prove this, it is only necessary to prove that (x, z, θ) verifies (11d'). From (11d) we have that (x, z, θ) verifies

$$\theta_j \geq \max_i \{C^i x - M(1 - \sum_{k \geq j} z_{ik})\}, \ j \in P \tag{5.17}$$

and for (11d'), we have to prove that (x, z, θ) also verifies

$$\theta_j \geq \max_i \{C^i x - M(\sum_{k<j} z_{ik})\}, \ j \in P. \tag{5.18}$$

We distinguish the following cases:

- If $\sum_{k \geq j} z_{i'k} = r > 1$ for some i' then

$$\theta_j \geq C^{i'} x + (r-1)M \geq \max_i \{C^i x - M(\sum_{k<j} z_{ik})\}, \tag{5.19}$$

  and the result holds.

- If $\sum_{k \geq j} z_{ik} = 1$ for all $i \in P$ then $\theta_j \geq \max_i \{C^i x\} \geq \max_i \{C^i x - M(\sum_{k<j} z_{ik})\}$ and the results is also proven.

- If $\sum_{k \geq j} z_{i'k} = 0$ for some i' then we distinguish to subcases. If $\sum_{k<j} z_{i'k} \geq 1$ then from (5.17) we easily get that (5.18) holds. Otherwise, $\sum_{k \in P} z_{i'k} = 0$ and by (5.11b) it does exist an i'' such that $\sum_{k \geq j} z_{i''k} = r > 1$. Thus, by using (5.19), equation (5.18) also holds.

□

Property 5.4- F^z and F_{R3}^z have the same set of optimal solutions.

Proof.

Since $\Omega^z \subset \Omega_{R2}^z$ and $\Omega_{R2}^z \subset \Omega_{R3}^z$ then $\Omega^z \subset \Omega_{R3}^z$ and it is enough to prove that any optimal solution to F_{R3}^z is feasible to F^z. Since the set of optimal solutions of F^z and F_{R2}^z coincide, we only need to prove that any optimal solution of F_{R3}^z is feasible for F_{R2}^z.

To see that any optimal solution (x, z, θ) to F^z_{R3} is feasible to F^z_{R2}, it is enough to see that $(x, z, \theta) \in \Omega^z_{R2}$, i.e. it satisfies inequalities (5.11b) and (11d).

By a similar argument to the one applied in Property 5.3, any optimal solution (x, z, θ) of F^z_{R3} satisfies $\sum_{j\in P} z_{ij} = 1$. Therefore, satisfying inequality (11d') implies inequality (11d).

Now (5.11b) follows directly from (5.11c) and (5.11b$_\leq$) since, otherwise, the sum of all constraints (5.11b$_\leq$) would not coincide with the sum of all constraints (5.11c).

To see that (x, z, θ) also satisfies (5.11b), let us suppose w.l.o.g. that there exists exactly one $j' \in P$ such that $\sum_{i\in P} z_{ij'} = 0$. Then, by adding up all constraints (5.11b$_\leq$) we have $p-1 \geq \sum_{j\in P}\sum_{i\in P} z_{ij} = \sum_{i\in P}\sum_{j\in P} z_{ij}$. Therefore, there must exist $i' \in P$ such that $\sum_{j\in P} z_{i'j} = 0$. Thus, we observe that we can construct $(x, \bar{z}, \bar{\theta})$, another optimal solution to F^z_{R3}, setting $\bar{z}_{ij} = z_{ij}$, if $i \neq i'$ and $\bar{z}_{i'k} = 1$ for any k. Clearly, $(x, \bar{z}, \bar{\theta})$ is a feasible solution to F^z_{R3} for some suitable $\bar{\theta}$, satisfying in addition

$$C^{i'}x \leq \bar{\theta}_k + M\sum_{\ell<k} z_{i'\ell}, \qquad \forall k \in P.$$

Therefore, this inequality allows for any $k \in P$ that $\bar{\theta}_k$ assumes a value smaller than or equal to θ_k, the one associated with the solution (x, z, θ), and therefore its objective value is at least as good as the previous one. Hence, $(x, \bar{z}, \bar{\theta})$ is also optimal. In addition, values $\bar{z}$ satisfy by construction that $\sum_{i\in P} \bar{z}_{ij'} = \sum_{i\neq i'} z_{ij'} + \bar{z}_{i'j'} = 0 + 1 = 1$. Therefore (5.11b) holds.

□

We can now relate the domains of the formulations considered so far.

Proposition 5.1- The following relationships hold

$$\Omega^z_0 \equiv \Omega^z \subsetneq \Omega^z_{R1} \subsetneq \Omega^z_{R2} \subsetneq \Omega^z_{R3}$$

`Proof.`

- $\Omega^z \subsetneq \Omega^z_{R1}$: Every feasible solution in Ω^z verifies inequalities of Ω^z_{R1}. However, a feasible solution in Ω^z_{R1} with $\theta_j \leq \theta_{j+1}$ for some $j \in P$ is not feasible in Ω^z.
- $\Omega^z_{R1} \subsetneq \Omega^z_{R2}$: Every feasible solution in Ω^z_{R1} verifies the inequalities of Ω^z_{R2}. However, a feasible solution in Ω^z_{R2} where for some $i \in P$, $z_{ij} = 1$, for all $j \in P$ is not feasible in Ω^z_{R1}.
- $\Omega^z_{R2} \subsetneq \Omega^z_{R3}$: Every feasible solution in Ω^z_{R2} verifies the inequalities of Ω^z_{R3}. However, a feasible solution in Ω^z_{R3} with $z_{ij} = 0$, $i, j \in P$ is not feasible in Ω^z_{R2}.

□

Proposition 5.2- The dimension of Ω_0^z is $p^2 - p + 1 + dim(Q)$.

Proof.

Suppose $Q \subseteq \mathbb{R}^n$. Then, Ω_0^z is embedded in a space of dimension p^2+p+n. Furthermore, since there are $2p-1$ linearly independent equations in (5.11b) and (5.11c) and the dimension of Q does not depend on relations (5.11b)-(5.11e), then the dimension of (5.11b)-(5.11f) is at most $p^2-p+1+dim(Q)$. Denote by $q = dim(Q)$ and by $\rho = p^2 - 2p + 1$. Next, we show that there exist $q + \rho + p + 1$ (equal to $p^2 - p + 2 + dim(Q)$) affinely independent points in Ω_0^z and consequently, the dimension of Ω_0^z is $p^2 - p + 1 + dim(Q)$.

Let $v = (v_j)_{j\in P}$ where $v_j = M + p - j + 1$ for $M > 0$ and sufficiently large. Denoting by $\mathbf{e^j} \in \mathbb{R}^p$ the j-th vector of the canonical basis in $\mathbb{R}^p$ and $0 < \varepsilon < 1$, let $\theta^j = \{v + \varepsilon\mathbf{e^j}, j \in P\}$. Moreover, let $\theta^{p+1} = (M, \dots, M)'$. We observe that the vectors θ^j, $j = 1, \dots, p+1$ are affinely independent and each one of them satisfies inequalities (5.11e).

Next, since $dim(Q) = q$, we take $q+1$ arbitrary affinely independent vectors $x^i \in Q$, $i = 1, \dots, q+1$. Furthermore, let $z^k \in \{0,1\}^{p^2}$ $k = 1, \dots, \rho+1$, be $\rho+1$ affinely independent vectors satisfying (5.11b) and (5.11c). Note that the latter is always possible since there are p^2 degrees of freedom for the coordinates of z variables and only $2p$ equations being one of them linearly dependent of the others.

Now, we prove that any point of the form $((x^i)', (z^k)', (\theta^l)')'$ $i = 1, \dots, q+1$, $k = 1, \dots, \rho+1$, $l = 1, \dots, p+1$ satisfies (5.11b)-(5.11e). Indeed, by construction the first block of coordinates defines a point in Q, the second block satisfies (5.11b) and (5.11c) and the third one (5.11e). Thus, it remains to prove that such a generic point also satisfies (11d) as follows:

$$C^i x^i \leq M \leq M + p - j + 1 \leq \theta_j^l + M(1 - z_{ij}^k), \quad \forall\, i, j.$$

Consider the $q + \rho + p$ points defined as the column vectors of the matrix $A = (A^1|A^2|A^3)$ where

$$A^1 = \begin{pmatrix} x^1 & x^2 & \dots & x^q \\ z^2 & z^1 & \dots & z^1 \\ \theta^2 & \theta^1 & \dots & \theta^1 \end{pmatrix}, \; A^2 = \begin{pmatrix} x^1 & x^1 & \dots & x^1 \\ z^1 & z^2 & \dots & z^\rho \\ \theta^2 & \theta^1 & \dots & \theta^1 \end{pmatrix}, \; A^3 = \begin{pmatrix} x^1 & x^1 & x^1 & \dots & x^1 \\ z^1 & z^3 & z^1 & \dots & z^1 \\ \theta^1 & \theta^2 & \theta^3 & \dots & \theta^p \end{pmatrix}.$$

By construction, each submatrix A^i has its column vectors linearly independent from one another since the i-th block is formed by linearly independent vectors. Next, clearly each column vector of A^1 is linearly independent from those of A^2 and A^3 and each column vector of A^2 is linearly independent from those of A^3. Therefore, the rank of A is $q + \rho + p = q + p^2 - p + 1$.

Finally, the column vectors of A are linearly independent and feasible points of (5.11b)-(5.11e). In addition, we can easily construct another feasible point, different from those considered previously and affinely independent from all of them, namely $((x^{q+1})', (z^{\rho+1})', (\theta^{p+1})')'$. Hence the dimension of Ω^z is $q + \rho + p = q + p^2 - p + 1$.

□

Proposition 5.3- The following inequalities define facets in Ω^z_0:

$$C^i x \leq \theta_p + M(1 - z_{ip}) \qquad i \in P \tag{5.20}$$

$$\theta_j \geq \theta_{j+1} \qquad j \in P : j < p \tag{5.21}$$

Proof.

(5.20) is a facet defining inequality:

We prove that for each $i' \in P$ there exist $dim(\Omega^z_0) = p^2 - p + dim(Q) + 1$ affinely independent points of Ω^z_0 that verify $C^{i'}x = \theta_p + M(1 - z_{i'p})$.

As in the proof of the above proposition, we take $q + 1$ arbitrary affinely independent points x^i, $i = 1, \ldots, q + 1$ in Q. Furthermore, let $z^k \in \{0, 1\}^{p^2}$ $k = 1, \ldots, \rho$, be ρ affinely independent points (recall that $\rho := p^2 - 2p + 1$) satisfying (5.11b), (5.11c) and $z_{i'p} = 1$. Note that the latter is always possible since there are p^2 degrees of freedom for the coordinates of z variables and $2p$ non redundant equations ($2p - 1$ as in the case above and $z_{i'p} = 1$).

Let $v^l = (v^l_j)_{j\in P}$ where $v^l_j = C^{i'}x^l + M + p - j$ if $j < p$ and $v^l_p = C^{i'}x^l$ for $M > 0$ and sufficiently large. Denoting by $\mathbf{e^j} \in \mathbb{R}^p$ the j-th vector of the canonical basis in $\mathbb{R}^p$ and $0 < \varepsilon < 1$, let $\bar{\theta}^{lj} = \{v^l + \varepsilon \mathbf{e^j}, j \in P\}$ if $j < p$ and $\bar{\theta}^{lp} = v^l$, $\bar{\theta}^{l,p+1} = (C^{i'}x^l + M, \ldots, C^{i'}x^l + M, C^{i'}x^l)'$. We observe that for each l fixed, the vectors $\bar{\theta}^{lj}$ $j = 1, \ldots, p + 1$ are affinely independent and each one of them satisfies inequalities (5.11e).

Now, we prove that any point of the form $((x^l)', (z^k)', (\theta^{lj})')'$ $k = 1, \ldots, \rho,$ $j = 1, \ldots, p + 1$ satisfies (5.11b)-(5.11e) and $z^k_{i'p} = 1$. Indeed, by construction the first block of coordinates defines a point in Q, the second block satisfies (5.11b), (5.11c) and $z_{i'p} = 1$, and the third one (5.11e). Thus, it remains to prove that such a generic point also satisfies (11d). We distinguish two cases:

- If $j < p$ then

$$C^i x^l \leq C^{i'}x^l + M + p - j + 1 + M = C^{i'}x^l + M + p - j + M(1 - z^k_{ij}) = \bar{\theta}^{lj} + M(1 - z^k_{ij}), \quad \forall i.$$

- If $j = p$ we have that

$$C^i x^l \leq C^{i'}x^l + M = C^{i'}x^l + M(1 - z^k_{ip}), \qquad \forall\, i \neq i',$$
$$C^{i'}x^l \leq C^{i'}x^l = C^{i'}x^l + M(1 - z^k_{i'p}), \quad \text{otherwise. (Recall that } z^k_{i'p} = 1.)$$

Consider the $q+\rho-1+p$ points defined as the column vectors of the matrix $\bar{A} = (\bar{A}^1|\bar{A}^2|\bar{A}^3)$ where

$$\bar{A}^1 = \begin{pmatrix} x^1 & x^2 & \dots & x^q \\ z^2 & z^1 & \dots & z^1 \\ \bar{\theta}^{11} & \bar{\theta}^{21} & \dots & \bar{\theta}^{q1} \end{pmatrix},\ \bar{A}^2 = \begin{pmatrix} x^1 & x^1 & \dots & x^1 \\ z^1 & z^2 & \dots & z^{\rho-1} \\ \bar{\theta}^{12} & \bar{\theta}^{11} & \dots & \bar{\theta}^{11} \end{pmatrix},\ \bar{A}^3 = \begin{pmatrix} x^1 & x^1 & x^1 & \dots & x^1 \\ z^1 & z^3 & z^1 & \dots & z^1 \\ \bar{\theta}^{11} & \bar{\theta}^{12} & \bar{\theta}^{13} & \dots & \bar{\theta}^{1p} \end{pmatrix}.$$

By construction, each submatrix $\bar{A}^i$ has its column vectors linearly independent from one another since the i-th block is formed by linearly independent vectors. Next, clearly each column vector of $\bar{A}^1$ is linearly independent from those of $\bar{A}^2$ and $\bar{A}^3$ and each column vector of $\bar{A}^2$ is linearly independent from those of $\bar{A}^3$. Therefore, the rank of A is $q+\rho-1+p = q+p^2-p$.

Finally, the column vectors of A together with the point $((x^{q+1})', (z^{\rho+1})', (\theta^{q+1,j})')'$ are feasible points of (5.11b)-(5.11e) that satisfy $C^{i'}x = \theta_p + M(1 - z_{i'p})$; and this last vector is clearly affinely independent from the those in $\bar{A}$, therefore (5.20) is a facet defining inequality for Ω^z.

(5.21) `is a facet defining inequality:`
In order to prove that for each $j' \in P \setminus \{p\}$ there exist $dim(\Omega_0^z) = p^2 - p + dim(Q) + 1$ affinely independent points of Ω_0^z that verify $\theta_{j'} = \theta_{j'+1}$, we can proceed analogously as before considering $v = (v_j)_{j=1}^p$, where $v_j = M + p - j + 1$ if $j \neq j'+1$ and $v_{j'+1} = M + p - j' + 2$ and the points $\hat{\theta}^j = \{v + \varepsilon(\mathbf{e^j} + \mathbf{e^{j'+1}}), j \in P \setminus \{p\}\}$. In addition, we take $\hat{\theta}^p = (M, \dots, M)'$. We observe that the vectors $\hat{\theta}^j$ $j = 1, \dots, p$ are affinely independent and each one of them satisfies $\hat{\theta}^j_{j'} = \hat{\theta}^j_{j'+1}$.

Any point of the form $((x^i)', (z^k)', (\hat{\theta}^l)')'$ $i = 1, \dots, q+1,\ k = 1, \dots, \rho+1,\ l = 1, \dots, p$ satisfies (5.11b)-(5.11e) and $\hat{\theta}^l_{j'} = \hat{\theta}^l_{j'+1}$.

Consider the $q+\rho+p-1$ points defined as the column vectors of the matrix $\hat{A} = (\hat{A}^1|\hat{A}^2|\hat{A}^3)$ where

$$\hat{A}^1 = \begin{pmatrix} x^1 & x^2 & \dots & x^q \\ z^2 & z^1 & \dots & z^1 \\ \hat{\theta}^2 & \hat{\theta}^1 & \dots & \hat{\theta}^1 \end{pmatrix},\ \hat{A}^2 = \begin{pmatrix} x^1 & x^1 & \dots & x^1 \\ z^1 & z^2 & \dots & z^{\rho} \\ \hat{\theta}^2 & \hat{\theta}^1 & \dots & \hat{\theta}^1 \end{pmatrix},\ \hat{A}^3 = \begin{pmatrix} x^1 & x^1 & x^1 & \dots & x^1 \\ z^1 & z^3 & z^1 & \dots & z^1 \\ \hat{\theta}^1 & \hat{\theta}^2 & \hat{\theta}^3 & \dots & \hat{\theta}^{p-1} \end{pmatrix}.$$

By construction, each submatrix $\hat{A}^i$ has its column vectors linearly independent from one another since the i-th block is formed by linearly independent vectors. Next, clearly each column vector of $\hat{A}^1$ is linearly independent from those of $\hat{A}^2$ and $\hat{A}^3$ and each column vector of $\hat{A}^2$ is linearly independent from those of $\hat{A}^3$. Therefore, the rank of $\hat{A}$ is $q+\rho+p-1 = q+p^2-p$.

Finally, the column vectors of $\hat{A}$ are linearly independent and are also feasible points of (5.11b)-(5.11e) that satisfy $\theta_{j'} = \theta_{j'+1}$. Next, we can easily add a new feasible point, for instance $((x^{q+1})', (z^{\rho+1})', (\hat{\theta}^p)')'$ that also satisfies $\theta_{j'} = \theta_{j'+1}$ and that is clearly affinely independent from the those in $\hat{A}$. Hence, (5.21) is a facet defining inequality for Ω^z.

□

Table 5.6 summarizes the previous proposed formulations. Formulas included on each formulation have been checked (✓) whereas those not appearing are marked with a dot (.).

	F_0^z	F^z	F_{R1}^z	F_{R2}^z	F_{R3}^z
$\min \sum_{j \in P} \omega_j \theta_j$	✓	✓	✓	✓	✓
$\sum_{i \in P} z_{ij} = 1,\ j \in P$	✓	✓	✓	✓	.
$\sum_{j \in P} z_{ij} = 1,\ i \in P$	✓	✓	✓	.	.
$\sum_{i \in P} z_{ij} \leq 1,\ j \in P$	.	.	.	.	✓
$C^i x \leq \theta_j + M(1 - z_{ij}),\ i,j \in P$	✓	.	.	.	.
$C^i x \leq \theta_j + M(1 - \sum_{k \geq j} z_{ik}),\ i,j \in P$	.	✓	✓	✓	.
$C^i x \leq \theta_j + M \sum_{k < j} z_{ik},\ i,j \in P$	.	.	.	.	✓
$\theta_j \geq \theta_{j+1},\ j \in P : j < p$	✓	✓	.	.	.
$x \in Q, z \in \{0,1\}^{p \times p}$	✓	✓	✓	✓	✓

Table 5.6: Summary of the proposed formulations for the OWAP.

5.3.3 OWAP formulations with variables for the values of cost functions occupying specific sorted positions

Another OWAP formulation can be obtained by defining an additional set of continuous variables $y = (y_{ij})_{i,j \in P} \in \mathbb{R}^{p \times p}$, where y_{ij} denotes the value of cost function i if it occupies the j-th position in the ordering. The formulation is as follows:

$$F_0^{zy}: \quad V = \min \sum_{j \in P} \omega_j \sum_{i \in P} y_{ij} \tag{5.22a}$$

$$s.t. \quad \sum_{i \in P} z_{ij} = 1 \qquad j \in P \tag{5.22b}$$

$$\sum_{j \in P} z_{ij} = 1 \qquad i \in P \tag{5.22c}$$

$$C^i x \leq \sum_{i' \in P} y_{i'j} + M(1 - z_{ij}) \qquad i,j \in P \tag{5.22d^0}$$

$$\sum_{i \in P} y_{ij} \geq \sum_{i \in P} y_{ij+1} \qquad j \in P : j < p \tag{5.22e}$$

$$x \in Q, z \in \{0,1\}^{p \times p} \tag{5.22f}$$

Next we study some properties of formulation F_0^{zy} and relate it to the OWAP formulations presented above. Denote by Ω_0^{zy} the domain of Problem F_0^{zy}. Consider first, for any $M > 0$ sufficiently large, the following set of inequalities

$$y_{ij} \leq M z_{ij}, \qquad i, j \in P. \tag{5.22g}$$

Property 5.5- There is an optimal solution to F_0^{zy} for which constraints (5.22g) hold.

`Proof.`
Observe that constraints (5.22d^0) imply that $\sum_{k\in P} y_{kj} \geq C^i x$ for all $i, j \in P$ with $z_{ij} = 1$. Since constraints (5.22b) indicate that for $j \in P$ fixed there exists a unique index, say $i(j)$ with $z_{i(j),j} = 1$, the above condition reduces to $\sum_{k\in P} y_{kj} \geq C^{i(j)} x$, for all $j \in P$. Because of the non-negativity of the cost coefficients, we can thus deduce that an optimal solution exists to F_0^{zy} in which

$$\sum_{k\in P} y_{kj} = C^{i(j)} x, \text{ for all } j \in P. \tag{5.23}$$

Let now $(x, y, z) \in \Omega_0^{zy}$ be such an optimal solution, and suppose it violates some constraint (5.22g). That is, there exist $i', j' \in P$ with $y_{i'j'} > M z_{i'j'}$. Hence, $\sum_{i\in P} y_{ij'} > M z_{i'j'}$, contradicting (5.23) unless $z_{i'j'} = 0$. In other words, $i(j') \neq i'$.

Consider now the solution $(x, \overline{y}, z)$, with the same x and z values as before where $\overline{y}$ is defined as follows:

$$\overline{y}_{ij} = \begin{cases} 0 & \text{if } i = i', \text{ and } j = j' \\ y_{i(j'),j'} + y_{i'j'} & \text{if } i = i(j'), \text{ and } j = j' \\ y_{ij} & \text{otherwise.} \end{cases}$$

Indeed $(x, \overline{y}, z) \in \Omega_0^{zy}$, as it is immediate to check that it satisfies constraints (5.22b)–(5.22f). Furthermore, by construction, it satisfies the constraint (5.22g) associated with i', j'. Finally, note that it is optimal to F_0^{zy}, since $\sum_{i\in P} \overline{y}_{ij} = \sum_{i\in P} y_{ij}$, for all $j \in P$. □

Note that if there is $j \in P$ with $\omega_j = 0$ then it is possible to have optimal solutions to F_0^{zy} that do not satisfy constraints (5.22g). However, because of Property 5.5, constraints (5.22g) can be useful to restrict the domain where optimal solutions are sought. Let

$$\Omega^{GS'} = \{(x, y, z, \theta) \text{ satisfying constraints (5.22b), (5.22c), (5.22d}^0\text{), (5.22e), (5.22f), (5.22g)}\}.$$

Then, a different formulation that also ensures to obtain an optimal solutions to F_0^{zy} is:

$$F^{GS'} \quad V = \min \sum_{j\in P} \omega_j \theta_j$$
$$s.t. \;\; (x, y, z, \theta) \in \Omega^{GS'}.$$

Formulation $F^{GS'}$ is closely related to the formulation used in Galand and Spanjaard (2012) for modeling the minimum cost spanning tree OWAP. In their formulation they operate on a domain which is like $\Omega^{GS'}$ except that constraints (5.22d^0) have been substituted by constraints

$$\sum_{j\in P} y_{ij} = C^i x \qquad i \in P. \tag{5.22h}$$

Let $\Omega^{GS} = \{(x, y, z, \theta)$ satisfying constraints (5.22b), (5.22c), (5.22e), (5.22f), (5.22g), (5.22h)$\}$, denote the domain used in Galand and Spanjaard (2012). Then, it is straightforward to conclude the following.

Property 5.6- The domains Ω^{GS} and $\Omega^{GS'}$ satisfy $\Omega^{GS} \subseteq \Omega^{GS'}$. Moreover, if (x^*, y^*, z^*) is an optimal solution of $F^{GS'}$ then it is also optimal for F^{GS} and conversely.

We can also relate F_0^{zy} with F_0^z and its variations. In particular, because of the relationship

$$\theta_j = \sum_{i\in P} y_{ij},\; j \in P. \tag{5.25}$$

we have:

Property 5.7- For each optimal solution to F_0^{zy}, $(x^*, y^*, z^*, \theta^*)$, there exists (x^*, z^*, θ^*) optimal solution for F_0^z and conversely. Moreover, $\sum_{j\in P} w_j \sum_{i\in P} y^*_{ij} = \sum_{j\in P} w_j \theta^*_j$.

By above result, we can derive variations of F^{zy} similar to the ones obtained for F^z with similar properties. These constructions are straightforward and therefore are left for the interested readers.

Table 5.7 summarizes the formulations proposed in this subsection that can be derived from those of Subsection 5.3.2. Constraints included in each formulation have been checked (✓) whereas those not appearing are marked with a dot (.).

5.3.4 Using variables defining relative positions of sorted cost function values

We close this section with another formulation which uses decision variables defining the relative positions of the sorted cost function values. As we have seen in Section 5.3.1 it is possible to describe permutations with variables representing the relative positions of the sorted values. Next we use such variables to obtain formulations for the OWAP.

	F_0^{zy}	F^{zy}	F_{R1}^{zy}	F_{R2}^{zy}	F_{R3}^{zy}
$\min \sum_{j\in P}\omega_j \sum_{i\in P} y_{ij}$	✓	✓	✓	✓	✓
$\sum_{i\in P} z_{ij} = 1,\ j \in P$	✓	✓	✓	✓	.
$\sum_{j\in P} z_{ij} = 1,\ i \in P$	✓	✓	✓	.	.
$\sum_{i\in P} z_{ij} \leq 1,\ j \in P$	.	.	.	.	✓
$C^i x \leq \sum_{i'\in P} y_{i'j} + M(1 - z_{ij}),\ i,j \in P$	✓	.	.	.	.
$C^i x \leq \sum_{i'\in P} y_{i'j} + M(1 - \sum_{k\geq j} z_{ik}),\ i,j \in P$	.	✓	✓	✓	.
$C^i x \leq \sum_{i'\in P} y_{i'j} + M\sum_{k<j} z_{ik},\ i,j \in P$	.	.	.	.	✓
$\sum_{i\in P} y_{ij} \geq \sum_{i\in P} y_{ij+1},\ j \in P : j < p$	✓	✓	.	.	.
$x \in Q, z \in \{0,1\}^{p\times p}$	✓	✓	✓	✓	✓

Table 5.7: Summary of the proposed formulations for the OWAP.

For $i, j \in P$, consider the binary variable $s_{ij},\ i, j \in P$ as

$$s_{ij} = \begin{cases} 1 & \text{if cost function } i \text{ is placed before position } j \text{ in the ordering,} \\ 0 & \text{otherwise.} \end{cases}$$

As we have seen in Section 5.3.1, for all $i, j \in P$, $s_{ij} = 1 - \sum_{k\geq j} z_{ik},\ i, j \in P$. Therefore, variables z and s are related by means of

$$z_{ij} = \begin{cases} s_{ij+1} - s_{ij} & i \in P, j = 1, ..., p-1 \\ 1 - s_{ij} & i \in P, j = p \end{cases} \tag{5.26}$$

Thus, we can reformulate the OWAP in the new space of the s variables as

$$F^s: \quad V = \min \sum_{j \in P} \omega_j \theta_j \tag{5.27a}$$

$$s.t. \quad \sum_{i \in P} s_{ij} = j - 1 \qquad j \in P \tag{5.27b}$$

$$s_{ij+1} - s_{ij} \geq 0 \qquad i, j \in P : j < p \tag{5.27c}$$

$$C^i x \leq \theta_j + M s_{ij} \qquad i, j \in P \tag{5.27d}$$

$$\theta_j \geq \theta_{j+1} \qquad j \in P : j < p \tag{5.27e}$$

$$x \in Q, s \in \{0, 1\}^{p \times p} \tag{5.27f}$$

Since F^s is obtained from F^z by a change of variable and there is a one to one correspondence between feasible solutions, we can state the following result. Let Ω^s be the feasible region of Problem F^s.

Property 5.8- For each solution $(x, s, \theta) \in \Omega^s$ there exists $(x, z, \theta) \in \Omega^z$ with equal objective value and conversely.

By analogy with the notation used in Section 5.3.2 let us define the following domains and problems related to F^s:

$$F^s_{R1} \quad V = \min \sum_{j \in P} \omega_j \theta_j$$

$$s.t. \quad (x, s, \theta) \in \Omega^s_{R1}.$$

with $\Omega^s_{R1} = \{(x, s, \theta)$ satisfying constraints (5.27b), (5.27c), (5.27d), (5.27f)$\}$.

$$F^s_{R2} \quad V = \min \sum_{j \in P} \omega_j \theta_j$$

$$s.t. \quad (x, s, \theta) \in \Omega^s_{R2}.$$

with $\Omega^s_{R2} = \{(x, s, \theta)$ satisfying constraints (5.27b), (5.27d), (5.27f)$\}$.

$$F^z_{R3} \quad V = \min \sum_{j \in P} \omega_j \theta_j$$

$$s.t. \quad (x, z, \theta) \in \Omega^z_{R3}.$$

with $\Omega^s_{R3} = \{(x, s, \theta)$ satisfying constraints (5.27b$_\leq$), (5.27d), (5.27f)$\}$, where (5.27b$_\leq$) are the inequality version of constraints (5.27b). That is,

$$\sum_{i \in P} s_{ij} \leq j - 1 \qquad j \in P. \tag{5.27b$_\leq$}$$

Property 5.9- The following relationships hold.

1. Every optimal solution to F^s_{R1} is optimal to F^s and conversely.
2. Every optimal solution to F^s_{R2} is optimal to F^s and conversely.
3. Every optimal solution to F^s_{R3} is optimal to F^s and conversely.
4. $\Omega^s \subsetneq \Omega^s_{R1} \subsetneq \Omega^s_{R2} \subsetneq \Omega^s_{R3}$.

`Proof.`

The proofs of the above statements follow directly from the relationship that links variables z and s, namely (5.9) and (5.10). Specifically, statement 1 follows from Property 5.2, statement 2 from Property 5.3, statement 3 from Property 5.4 and statement 4 from Property 5.1.

□

5.3.5 Formulations summary

Table 5.8 summarizes the previous formulations in this subsection. Constraints included in each formulation have been checked (✓) whereas those not appearing are marked with a dot (.).

	F^s	F^s_{R1}	F^s_{R2}	F^s_{R3}
$\min \sum_{j \in P} \omega_j \theta_j$	✓	✓	✓	✓
$\sum_{i \in P} s_{ij} = j - 1, j \in P$	✓	✓	✓	.
$s_{ij+1} - s_{ij} \geq 0, i, j \in P : j < p$	✓	✓	.	.
$\sum_{i \in P} s_{ij} \leq j - 1, j \in P$	.	.	.	✓
$C^i x \leq \theta_j + M s_{ij},\ i, j \in P$	✓	✓	✓	✓
$\theta_j \geq \theta_{j+1},\ j \in P : j < p$	✓	.	.	.
$x \in Q, z \in \{0,1\}^{p \times p}$	✓	✓	✓	✓

Table 5.8: Summary of the proposed formulations for the OWAP.

5.4 Valid inequalities and reinforcements for the OWAP formulation

5.4.1 Valid inequalities for the (OWAP) formulation

In this section we derive different valid inequalities for all the formulations presented in previous sections. For the sake of simplicity, we present all inequalities for the formulations developed in Subsection 5.3.2. However, all these inequalities can be easily adapted to the remaining formulations just by means of the substitutions explained by Equations (5.10) and (5.25).

- *Constraints related to bounds of cost function values.* Let l_i (u_i) denote the minimum (maximum) objective value relative to cost function $i \in P$, respectively. It is clear that l_i (u_i) are valid lower (upper) bounds on the value of objective i, independently of the position of cost function i in the ordering. Therefore we obtain the following two sets of constraints which are valid for the OWAP:

$$l_i \leq C^i x \leq u_i \qquad i \in P \tag{5.31}$$

- *Constraints related to bounds of values in specific positions.* Let l_j^π (u_j^π) denote the j-th lowest (largest) value of l_i (u_i). Then, l_j^π (u_j^π) is a valid lower (upper) bound of the objective function sorted in position j, that is

$$l_j^\pi \leq \theta_j \leq u_j^\pi \qquad j \in P \tag{5.32}$$

- *Constraints related to bounds of cost function values in specific positions.* Let l_{ij} and u_{ij} denote valid lower and upper bounds on the value of objective i if it occupies position j, respectively. Then, lower and upper bounds on the value of objective i are

$$\min_{j\in P} l_{ij} \leq C^i x \leq \max_{j\in P} u_{ij} \qquad i \in P \tag{5.33}$$

  Analogously to (5.32), we can sort the j-th lowest (largest) value of $\min_{j\in P} l_{ij}$ obtaining the following inequality

$$\min_{i\in P} l_{ij} \leq \theta_j \leq \max_{i\in P} u_{ij} \qquad j \in P \tag{5.34}$$

- There are also different bounds on the value of the cost function i and the value of the cost function sorted in position j:

$$\sum_{j\in P} \max\{l_i, l_j^\pi\} z_{ij} \leq C^i x \leq \sum_{j\in P} \min\{u_i, u_j^\pi\} z_{ij} \qquad i \in P \tag{5.35}$$

$$\sum_{i \in P} \max\{l_i, l^\pi_j\} z_{ij} \leq \theta_j \leq \sum_{i \in P} \min\{u_i, u^\pi_j\} z_{ij} \qquad j \in P \qquad (5.36)$$

- The inclusion of the following constraint also allows to consider, in the original formulations in Section 5.3, weights $\omega \in \mathbb{R}$ that, consequently, could take both negative and positive values.

$$\theta_j \leq \max_{i \in P}\{u_{ij}, C^i x + M(1 - z_{ij})\} \qquad i, j \in P \qquad (5.37)$$

- *Constraints related to positions in the ordering.* Constraints (5.38) impose that the position values are ordered in non-increasing order.

$$\theta_j \geq \theta_{j+1} \qquad j \in P \backslash \{p\} \qquad (5.38)$$

- *Constraints related to subsets of cost functions.* Next, we observe that for any subset $I \subseteq P$, of size $k = 1, ..., p$

$$\sum_{i \in I} C^i x \leq \sum_{j=1}^{k} \theta_j \qquad I \subseteq P \qquad (5.39)$$

In particular, we consider the cases when $I = \{i\}$, $I = \{i, i' \in P\}$, $I = P \setminus \{i\}$ and $I = P$.

5.4.2 Valid inequalities for the (OWAP2) formulation

Note first that all previous inequalities from Section 5.4.1 can be applied to the two-index formulation of the OWAP substituting $\theta_j = \sum_{i \in P} y_{ij}$. Additionally, the following inequalities provide a reinforcement to the formulations using y variables:

- The following inequality combined with (5.22e) improves considerably the LP relaxation of the OWAP

$$\sum_{k \in P} y_{ik} = C^i x \qquad i \in P \qquad (5.40)$$

- Constraint (5.22e) can be disaggregated by $j \in P$ as:

$$y_{ij} \leq \sum_{i' \in P} y_{i'j} + \min\{u_i, u^\pi_j\}(1 - \sum_{k \geq j} z_{ik}) \qquad i, j \in P \qquad (5.41)$$

- We can also establish a lower bound on the value of cost function $i \in P$ if it is ordered in position $j \in P$ by relating the x, y and z variables as follows:

$$C^i x \leq y_{ij} + u^\pi_j (1 - z_{ij}) \qquad i, j \in P \qquad (5.42)$$

Observe that, for i, j fixed, the above constraint imposes a lower bound on the value y_{ij} only when cost function $i \in P$ is ordered in position $j \in P$, and becomes inactive otherwise.

- We can also relate the values of two different cost functions between them, depending on their positions. In particular,

$$\sum_{k \geq j+1} y_{ik} \leq y_{i'j} + u_i(1 - z_{i'j} - z_{ij}) \qquad i, i', j \in P, i \neq i', j \neq p \tag{5.43}$$

For i, i', j fixed, constraint (5.43) establishes that when cost function i' occupies position j, its value cannot be smaller than that of cost function i, provided that cost function i is ordered after j. Observe that the constraint becomes inactive when i is ordered before j or is in j position (since in this case $\sum_{k \geq j+1} y_{ik} = 0$) and when i' does not occupy position j.

- A better effectiveness of the previous inequalities can be obtained by means of

$$y_{ij+1} \leq y_{i'j} + (1 - z_{ij+1})u_{ij+1} + (1 - z_{i'j})u_{i'j} \qquad i, i', j \in P, i \neq i', j \neq p \tag{5.44}$$

which can be further reinforced to

$$y_{ij+1} \leq y_{i'j} + (1 - z_{ij+1}) \min\{u_i, u^{\pi}_{j+1}\} + (1 - z_{i'j}) \min\{u_{i'}, u^{\pi}_{j}\} \quad i, i', j \in P, i \neq i', j \neq p. \tag{5.45}$$

5.4.3 Lower and upper bounds: Elimination tests

Several of the inequalities presented above use valid lower and upper bounds on the values of the different cost functions, l_i and u_i, respectively. As mentioned above, the minimum and maximum objective value with respect to each cost function provide such bounds. However, tighter bounds can be very useful for obtaining tighter constraints. One possibility is to use lower and upper bounds on the value of each objective for the different positions in the ordering. In particular, if L_{ij} and U_{ij} denote valid lower and upper bounds on the value of objective i if it occupies position j, respectively, then lower and upper bounds on the value of objective i are $l_i = \min_{j \in P} L_{ij}$ and $u_i = \max_{j \in P} U_{ij}$, respectively. For $i, j \in P$ given, L_{ij} and U_{ij} can be obtained in different ways. One alternative is to solve the linear programming (LP) relaxation of the formulation, both for the minimization and the maximization of cost function i, with the additional constraint that it occupies position j. In this case L_{ij} (U_{ij}) is the optimal value of the minimization (maximization) OWAP problem in which we fix the ordering variable at value 1, i.e. $z_{ij} = 1$.

Next we present simple tests which can help to eliminate some variables by fixing their values. Broadly speaking these tests compare the value of a lower bound associated with the decision of setting (or not setting) objective i at position j with the value of a known upper bound. If the value of the lower bound exceeds the value of the upper bound, the associated decision variable can be fixed. Any feasible solution yields a valid upper bound, which corresponds to its value with respect to the objective function. In the following we use U to denote the value of the upper bound corresponding to the best-known solution. We also denote by L^0_{ij} the optimal value of the minimization OWAP problem in which we fix the ordering variable at value 0, i.e. $z_{ij} = 0$. Then for each $i \in P$, $j \in P$ we have

- If $L_{ij} > U$ then $z_{ij} = 0$ (no optimal solution will have objective i in position j).

- If $L^0_{ij} > U$ then $z_{ij} = 1$ (no optimal solution will not have objective i in position j).

5.5 The OWA problem on shortest paths and minimum cost perfect matchings

This section presents the formulations of the combinatorial objects that we use in our computational experiments, namely shortest paths and minimum cost perfect matchings. In order to test our results we have chosen two of the most well-known formulations for these two problems. These formulations have to be combined with those presented in previous sections to provide valid OWAP models for the Shortest Path Problem (SPP) (see, e.g., Cherkassky et al., 1996; Ramaswamy et al., 2005) and the Perfect Matching Problem (PMP) (see, e.g., Edmonds, 1965; Grötschel and Holland, 1985). All the details are given in what follows.

5.5.1 The shortest path problem

We consider now the $OWAP$ when Q is the feasible set of the SPP (see, e.g., Cherkassky et al., 1996). Let $G = (V, E)$ be an undirected graph with set of vertices V, $|V| = n$ and set of edges E, $|E| = m$. In addition to the sets of variables required to model the order of the p cost functions ranked by non-increasing criterion values, we will need additional variables used to model the structure of the combinatorial object (shortest path in this case). For modeling the shortest path between two selected vertices, $u_1, u_n \in V$ we use a flow-based formulation, in which binary design variables x are related to continuous flow variables φ. In particular, for each $e = (u, v) \in E$ let

$$x_e \equiv x_{uv} = \begin{cases} 1 & \text{edge } e \equiv (u, v) \text{ is in the shortest path,} \\ 0 & \text{otherwise.} \end{cases}$$

As usual, paths between $u_1, u_n \in V$ can be obtained by identifying the arcs that are used when one unit of flow is sent from u_1 to u_n. For the flow variables we consider a directed network, with set of vertices V and set of arcs A which contains two arcs, one in each direction, associated with each edge of E. For each $(u, v) \in A$ we define the decision variables φ_{uv} which represents the amount of flow through arc (u, v). Then a characterization of the domain of feasible solutions (Q) for the SPP is:

$$\sum_{(u,v)\in A} \varphi_{uv} - \sum_{(u,v)\in A} \varphi_{vu} = 1 \qquad u = u_1 \tag{5.46a}$$

$$\sum_{(u,v)\in A} \varphi_{uv} - \sum_{(u,v)\in A} \varphi_{vu} = -1 \qquad u = u_n \tag{5.46b}$$

$$\sum_{(u,v)\in A} \varphi_{uv} - \sum_{(u,v)\in A} \varphi_{vu} = 0 \qquad u \in V \setminus \{u_1, u_n\} \tag{5.46c}$$

$$\varphi_{uv} + \varphi_{vu} \le x_{uv} \qquad (u, v) \in E \tag{5.46d}$$

$$\varphi_{uv} \ge 0 \qquad (u, v) \in A \tag{5.46e}$$

$$x_e \in \{0, 1\} \qquad e \in E \tag{5.46f}$$

Constraints (5.46a)–(5.46c) guarantee flow conservation at any vertex of the network. Constraints (5.46d) relate the φ and x variables, by imposing that all the edges used for sending flow in some direction are activated.

5.5.2 The perfect matching problem

We consider now the $OWAP$ when Q is the feasible set of the PMP (see, e.g., Edmonds, 1965). It is well known that the PMP is polynomially solvable by using the Blossom algorithm (Edmonds, 1965). However, to the best of our knowledge it is not known how such an algorithm could be used for solving an OWAP in which Q is given by the set of perfect matchings on a given graph. Indeed, this can be done by using any of the OWAP formulations we have introduced in the previous sections.

Let $G = (V, E)$ be an undirected graph with set of vertices V, $|V| = n$ and set of edges E, $|E| = m$. In addition to the sets of variables required to model the order of the p cost functions ranked by non-increasing criterion values, we will need additional variables used to model the structure of the combinatorial object (perfect matching in this case). For modeling the perfect matching we use binary design variables x associated with the edges of the graph. In particular, for each $e = (u, v) \in E$ let

$$x_e \equiv x_{uv} = \begin{cases} 1 & \text{edge } e \equiv (u, v) \text{ is in the matching,} \\ 0 & \text{otherwise.} \end{cases}$$

We introduce some additional notation. For $S \subset V$, $E(S) = \{e = (u, v) \in E \mid u, v \in S\}$ and $\delta(S) = \{e = (u, v) \in E \mid u \in S, v \notin S\}$. When S is a singleton, i.e. $S = \{u\}$ with $u \in V$ we simply write $\delta(\{u\}) = \delta(u)$. Then, a characterization of the domain of feasible solutions for the PMP (Q) is:

$$\sum_{e \in \delta(u)} x_e = 1 \qquad u \in V \tag{5.47a}$$

$$x_e \in \{0,1\} \qquad e \in E \tag{5.47b}$$

Constraints (5.47a) guarantee that in the solution the degree of every vertex is one.

5.5.3 Complexity

In this section we prove that the OWAP combined with the PMP (OWAPMP) is NP-complete even on bipartite graphs. The reader may note that a similar argument that the one presented can be used to prove the NP-completeness of the OWAP when Q is the SPP.

Given a graph $G = (V, E)$ with weights (c_e^1, c_e^2) for each $e \in E$, $\omega-$weights $\omega_1, \omega_2 \geq 0$ and a constant K, the decision version of the OWAPMP is the following: Is there a matching $\mathcal{M}$ of G such that $\omega_1\theta_1(\mathcal{M}) + \omega_2\theta_2(\mathcal{M}) \leq K$? Reduction comes from the Partition with Disjoint Pairs (PDP) problem: Given n pairs of integers (a_i, b_i), $i \in \{1, ..., n\}$ with $\sum_i (a_i + b_i) = Q$, is there $S \subset \{1, 2, ..., n\}$ such that $\sum_{i \in S} a_i + \sum_{i \notin S} b_i = Q/2$?

Assume a solution S of (PDP) exists. Then we construct a bipartite graph with $2n$ vertices $\{1, 2, ..., 2n\}$ such that $c_{ii+n} = (a_i, b_i)$ and $c_{ij} = (b_i, a_i)$ $\forall j \neq i + n$, $j \geq n + 1$. We take $\omega_1 > 0$, $\omega_2 = \omega_1 - \epsilon > 0$ and $K = Q(\omega_1 - \epsilon/2)$. For $i \in S$ $(i, n + i) \in \mathcal{M}$, and if $i \notin S$ $(i, j(i)) \in \mathcal{M}$, $j \neq i$, $i, j \notin S$. That is, we can choose $j(i)$ for each $i \notin S$ in such a way that the above construction is a matching. It is clear that

$$\theta_1(\mathcal{M}) = \sum_{i \in S} a_i + \sum_{i \notin S} b_i = Q/2$$

and this implies that $\theta_2(\mathcal{M}) = Q/2$ and therefore a solution for OWA-matching with $K = Q(\omega_1 - \epsilon/2)$ exists.

Conversely, if we have a solution $\mathcal{M}$ to OWAPMP less than or equal to $Q(\omega_1 - \epsilon/2)$, there must be a subset of nodes $i \in S$ for which the edge is $(i, i + n)$ and for the remaining we take $(i, j(i))$ as above with costs (b_i, a_i). If not all nodes of S go with the cost (a_i, b_i) one of the two objective functions has a value $Q/2 + \Delta$ and the other $Q/2 - \Delta$ for some $\Delta > 0$. Therefore $\theta_1(\mathcal{M}) = Q/2 + \Delta$, $\theta_2(\mathcal{M}) = Q/2 - \Delta$ and

$$\omega_1\theta_1(\mathcal{M}) + (\omega_1 - \epsilon)\theta_2(\mathcal{M}) = \omega_1(Q/2 + \Delta) + (\lambda_1 - \epsilon)(Q/2 - \Delta) = Q(\omega_1 - \epsilon/2) + \epsilon\Delta.$$

5.6 Computational experience

In this section we report on the results of some computational experiments we have run, in order to compare empirically the proposed formulations and reinforcements. We have studied the OWAP over the two combinatorial objects proposed: Shortest Paths and Minimum Cost Perfect Matchings. The best formulation obtained for each combinatorial object, has been later used for studying the proposed valid inequalities, including them one by one separately. Accordingly, for each combinatorial object, we have obtained results for 14 basic formulations (i.e., without adding any valid inequality) plus 19 "reinforced" formulations. For the sake of readability, we display results in tables just for the three best basic formulations and graphics for both basic and reinforced formulations. For further details, the reader may refer to Fernández et al. (2013) in order to check all the results obtained in the computational experiments organized by tables.

The OWA operator allows to model various aggregation functions according to the vector of weights w (see, e.g., Ogryczak and Olender, 2012). Some examples are the minimum, maximum, median, center or k-centrum functions. Therefore, the variation of w into non/monotonic or non/symmetric weights is directly connected with a problem structure and thus with a problem complexity (Kasperski and Zielinski, 2013). Some elegant linearization of OWA functions have been proposed in the literature for some subclasses of OWA operators (see, e.g., Ogryczak and Sliwinski, 2003; Ogryczak and Tamir, 2003, for convex OWA with decreasing weights). For keeping the length of the chapter within some reasonable limits, in our computational experience we study a particular case of the OWA operator, namely the Hurwicz criterion (Hurwicz, 1951), defined as $\alpha \max_{i \in P} y_i + (1-\alpha) \min_{i \in P} y_i$. This criterion is non-monotonic and non-convex and, in our experience, its behavior in terms of computational effort to get optimal solutions is similar to that of other non-convex OWA criteria. In addition, this objective has been already considered when analyzing the behavior of OWA operators in multiobjective optimization (see, e.g., Galand and Spanjaard, 2012) and it is of special interest for being non-convex since the sorting weights, α, are not in non-increasing order (Grzybowski et al., 2011, Puerto and Tamir, 2005). The considered values of α are $\{0.4, 0.6, 0.8\}$ and the number of objectives ranges in $|P| \in \{4, 7, 10\}$. Graphs generation is described below considering three different sizes of the graph according to $|V| \in \{100, 225, 400\}$. In addition, for each selection of the parameters $(|V|, p, \alpha)$, 10 instances were randomly generated so, in total, we have a set of 270 benchmark instances. All instances were solved with the MIP Xpress optimizer, under a Windows 7 environment in an Intel(R) Core(TM)i7 CPU 2.93 GHz processor and 8 GB RAM. Default values were used for all solver parameters. A CPU time limit of 600 seconds was set.

For the benchmark instances, we generated square grid networks produced as with the SPGRID generator of Cherkassky et al. (1996) for both combinatorial objects. Nodes of these graphs correspond to points on the plane with integer coordinates $[x, y]$, $1 \leq x \leq \sqrt{|V|}$, $1 \leq y \leq \sqrt{|V|}$. These points are connected "forward" by arcs of the form $([x, y], [x+1, y])$, $1 \leq x < \sqrt{|V|}$, $1 \leq y \leq \sqrt{|V|}$; "up" by arcs of the form $([x, y], [x, y+1])$, $1 \leq x \leq \sqrt{|V|}$, $1 \leq y < \sqrt{|V|}$ and "down" by arcs of the form $([x, y], [x, y-1])$, $1 \leq x \leq \sqrt{|V|}$, $1 < y \leq \sqrt{|V|}$ and by arcs of the form $([x, y], [x+1, y-1])$, $1 \leq x \leq \sqrt{|V|}$, $1 < y \leq \sqrt{|V|}$. The components of the cost vectors are randomly drawn from a uniform distribution on $[1, 100]$. Note also that shortest paths are computed between nodes 1 and $|V|$ whereas node $|V|$ is removed for the PMP when $|V|$ is odd.

Each of our tables reports the following items. Each row corresponds to a group of 10 instances with the same characteristics ($|V|, p, \alpha$) indicated in the first three columns. Column $t(\#)$ reports firstly the average running time in seconds of the 10 instances of the row. In addition, if at least one instance reaches the CPU time limit, we indicate in brackets the number of instances that could be solved to optimality within the maximum CPU time limit and, in such a case, we compute the average running time by using the CPU time limit for those instances that could not be solved to optimality. Column t^*/gap^* reports the biggest CPU time over the 10 instances of the group. Whenever the time limit is reached, the relative gap (indicated with a percentage %) is reported instead. Column $\#nodes$ indicates the average number of nodes explored in the branch and bound tree and column gap_{LR} reports the relative gap computed with the best solution found by the solver and the linear relaxation optima at the root node. All tables report analogous items for the different formulations described along the chapter. The best three formulations for each combinatorial object are F^z_{R2}, F^{zy}_{R2}, and F^s for the SPP; and F^z_{R1}, F^{zy}_{R1}, and F^s_{R1} for the PMP. Entries in bold remark best values among the 16 basic formulations (all tables are available at Fernández et al., 2013).

Figures 5.1 and 5.2 summarize the comparative results of all proposed basic formulations applied to each combinatorial object respectively. In these graphics the $x-$axis displays the different variations of the formulations presented in Section 5.3 and the $y-$axis the features analyzed. All displayed bars represent percentages of mean values computed over 90 instances with $|V| = 400$. These are the 90 hardest instances for the solver among the 270 we generated.

In particular the row labeled with "t, gap" shows a bar with the mean values of the running times measured in percentage over 600 seconds. For those instances reaching the time limit, we compute the mean running time taking the value of the time limit. Moreover, a dashed line indicates the percentage of worst case gap among those instances that have reached the time limit. The columns in the row labeled with "*nodes*" show the percentage of nodes over 10^6 that have been visited in the branch and bound tree. The columns in the row labeled with "gap_{LR}" report the percentage gap relative to the best solution found by the solver and the linear relaxation optima at the root node.

From the results displayed in Table 5.9 and Figure 5.1, we observe first that the gap_{LR} is similar for all formulations except for F^z_0 and F^{zy}_0, where a 100% of gap is reached. Formulations F^z_{R2} and F^{zy}_{R2} increase slightly the gap_{LR} in comparison with the remaining formulations but this does not affect negatively in the exploration as we see next. The values of *nodes* and t, gap are strongly related for each one of the formulations. F^z_0, F^z_{R3}, F^{zy}_0 and F^{zy}_{R3} give the worst values. In contrast, F^z_{R2}, F^{zy}_{R2} and F^s produce the best values. In addition, we observe a regular behavior among all formulations with s variables, namely F^s, F^s_{R1}, F^s_{R2} and F^s_{R3}. Regarding to the PMP, analogous conclusions can be obtained in Table 5.10 and Figure 5.2 for the gap_{LR} and the relations between *nodes* and t, gap. However, in this case, formulations F^z_{R1} and F^{zy}_{R1} produce the best values together with F^s_{R1}, F^s_{R2} and F^s_{R3}.

Figures 5.3 and 5.4 report analogous items as Figures 5.1 and 5.2, but now when the valid inequalities of

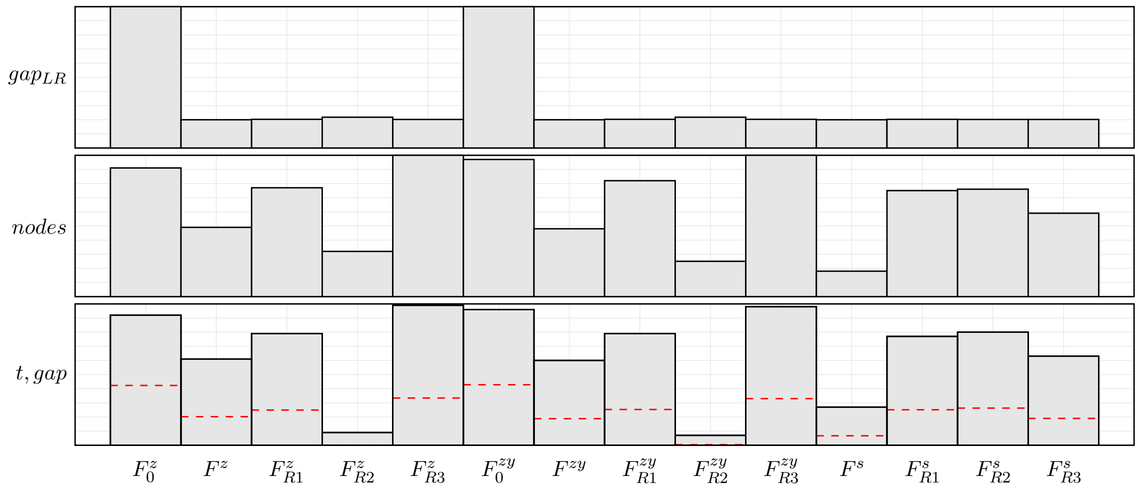


Figure 5.1: Comparative results for the proposed OWAP basic formulations applied to the Shortest Path Problem ($p = 10$, $|V| = 400$).

Inst			F_{R2}^z				F_{R2}^{zy}				F^s			
$\|V\|$	p	α	$t(\#)$	t^*/gap^*	$\#nodes$	gap_{LR}	$t(\#)$	t^*/gap^*	$\#nodes$	gap_{LR}	$t(\#)$	t^*/gap^*	$\#nodes$	gap_{LR}
100	4	0.4	0.5	**0.6**	15	55.79	**0.4**	**0.6**	**13**	55.79	8.4	59.9	16959	**53.72**
100	4	0.6	**0.5**	**0.6**	**41**	40.17	12	116.5	56370	40.17	31.9	213.4	113492	**37.39**
100	4	0.8	**0.4**	**0.5**	61	24.26	**0.4**	**0.5**	**48**	24.26	2.4	7.9	2871	**20.79**
100	7	0.4	**0.6**	**0.7**	200	52.77	**0.6**	0.8	**177**	52.77	121 (8)	40.66%	210086	**51.11**
100	7	0.6	**0.7**	**0.8**	**360**	38.19	0.8	1.6	468	38.19	121 (8)	22.74%	193167	**36.01**
100	7	0.8	**0.9**	1.6	839	23.76	**0.9**	**1.4**	**760**	23.76	20.8	125.9	36870	**21.08**
100	10	0.4	**2.1**	**4.2**	**6658**	51.98	2.5	10.3	9239	51.98	195.2 (7)	43.64%	273035	**49.29**
100	10	0.6	4.1	13.2	16386	37.89	**2.8**	**11.1**	**9985**	37.89	178.6 (8)	24.44%	238513	**34.4**
100	10	0.8	5.5	27.9	23353	24.83	13.1	49.4	57599	24.83	95.3	500.7	127230	**20.61**
225	4	0.4	**0.8**	**1**	48	55.77	**0.8**	1.1	**45**	55.77	64.4 (9)	52.43%	29874	55
225	4	0.6	**0.8**	**1**	**44**	39.42	**0.8**	**1**	49	39.42	91.5 (9)	31.77%	41747	**38.31**
225	4	0.8	**0.8**	**1.1**	95	22.13	**0.8**	1.2	**84**	22.13	243.8 (6)	14.08%	70842	**20.7**
225	7	0.4	**1.2**	**1.3**	**99**	52.61	1.3	1.8	151	52.61	129 (8)	49.52%	41763	51.29
225	7	0.6	**3.3**	**8.8**	**1554**	37.63	16.2	143.6	10871	37.63	185.6 (7)	31.25%	63146	35.83
225	7	0.8	4.6	22.1	3082	22.76	**2.6**	**6.1**	**1204**	22.76	305 (5)	14.19%	105127	**20.44**
225	10	0.4	9.1	62.7	6427	51.68	**5.4**	**24.9**	**4222**	51.68	317.1 (5)	49.98%	95076	50.33
225	10	0.6	15.2	56.6	10148	37.07	**10.8**	**39.7**	**7537**	37.07	319.5 (5)	32.14%	96370	**35.15**
225	10	0.8	38.1	147.8	41223	23.12	**29.6**	**141.5**	**32090**	23.12	279.6 (6)	15.16%	85419	20.81
400	4	0.4	1.4	1.8	57	55.07	**1.3**	**1.6**	**55**	55.07	3.3	16.8	286	**54.44**
400	4	0.6	1.6	2	95	38.71	**1.5**	**1.8**	**76**	38.71	88.5 (9)	35.79%	13806	**37.84**
400	4	0.8	**1.8**	**2.9**	**182**	21.57	**1.8**	3.1	265	21.57	255.9 (6)	17.21%	49806	**20.47**
400	7	0.4	**6.5**	**41.1**	**1102**	52.72	19.3	169	4370	52.72	76.4 (9)	50.67%	9192	**51.85**
400	7	0.6	**9.4**	**62.6**	**2952**	37.41	63.4 (9)	33.32%	10416	37.48	70.9 (9)	34.59%	8711	**36.27**
400	7	0.8	8.1	30.2	**1994**	21.87	**7.2**	**24.6**	1999	21.87	368.8 (4)	18.29%	32614	20.41
400	10	0.4	158.5 (9)	1.09%	100979	51.8	**116.1**	**242.7**	83991	51.8	306.4 (5)	48.93%	24184	**50.73**
400	10	0.6	61.8	121.5	37448	36.48	**33.2**	**115.9**	17308	36.48	132.4 (8)	31.48%	**10395**	**35.01**
400	10	0.8	229.9 (8)	0.61%	143034	21.82	155.6 (**9**)	**0.04%**	104042	21.82	**142.6** (8)	17.18%	12829	19.99

Table 5.9: Results obtained for the three best OWAP basic formulations applied to the Shortest Path Problem

Section 5.4 are incorporated into the best basic formulations obtained for each combinatorial object. The $x-$axis displays the different variations in the formulations, starting first with the best basic formulation. Next labels refer to the valid inequality that has been added. Labels of the valid inequalities correspond with those of Section 5.4, where ".1" and ".2" refer to the two inequalities displayed in a single equation (for example the two valid inequalities of equation (5.31) are labeled as (5.31.1) and (5.31.2)). In the following we will refer indistinctly to a valid inequality and the

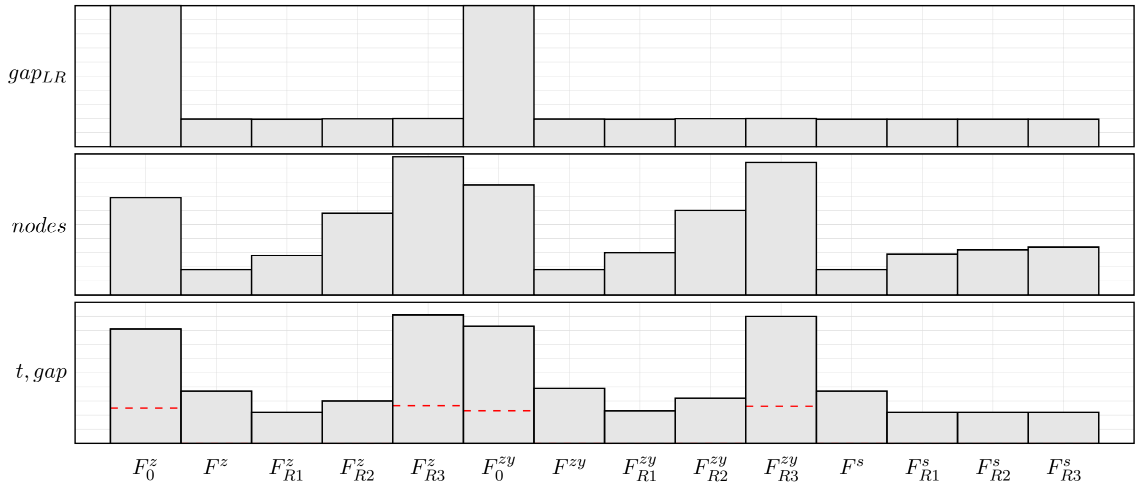


Figure 5.2: Comparative results for the proposed OWAP basic formulations applied to the Perfect Matching Problem ($p = 10$, $|V| = 400$).

Inst			F_{R1}^z				F_{R1}^{zy}				F_{R1}^s			
$\|V\|$	p	α	$t(\#)$	t^*/gap^*	$\#nodes$	gap_{LR}	$t(\#)$	t^*/gap^*	$\#nodes$	gap_{LR}	$t(\#)$	t^*/gap^*	$\#nodes$	gap_{LR}
100	4	0.4	0.6	**0.7**	186	**55.44**	0.6	**0.7**	139	**55.44**	**0.5**	**0.7**	147	**55.44**
100	4	0.6	**0.6**	0.7	152	**38.98**	**0.6**	0.7	140	**38.98**	**0.6**	0.8	175	**38.98**
100	4	0.8	**0.6**	**0.7**	302	**21.53**	0.7	0.8	329	**21.53**	**0.6**	**0.7**	157	**21.53**
100	7	0.4	1	1.3	236	**52.18**	1	**1.2**	256	**52.18**	1	**1.2**	**205**	**52.18**
100	7	0.6	**1.1**	1.4	480	**35.97**	1.2	1.8	591	**35.97**	1.2	1.6	529	**35.97**
100	7	0.8	1.4	2	965	**20.27**	1.5	2.3	1008	**20.27**	**1.3**	**1.8**	1075	**20.27**
100	10	0.4	**1.5**	1.9	299	**50.66**	1.7	4.1	580	**50.66**	**1.5**	1.9	333	**50.66**
100	10	0.6	**1.9**	**2.6**	963	**34.85**	**1.9**	2.9	985	**34.85**	2	2.8	922	**34.85**
100	10	0.8	6	19.4	6329	**20.2**	5.6	17.6	**5364**	**20.2**	6.1	19.8	7018	**20.2**
225	4	0.4	2.1	4.4	1188	**55.09**	2	**2.9**	990	**55.09**	**1.9**	4.1	1095	**55.09**
225	4	0.6	**1.7**	2.9	1236	**38.57**	**1.7**	2.5	1239	**38.57**	**1.7**	**2.2**	**982**	**38.57**
225	4	0.8	**1.9**	3.2	1101	**21.09**	**1.9**	3.7	1240	**21.09**	2	3.6	1221	**21.09**
225	7	0.4	**7.1**	**22.8**	9208	**52.34**	8.4	36	5617	**52.34**	8.7	29.3	6308	**52.34**
225	7	0.6	10	16	6038	**36.27**	9.7	18.3	6206	**36.27**	**8.8**	**15.9**	**5432**	**36.27**
225	7	0.8	17.2	62.5	10491	**20.32**	17.1	**48.7**	10746	**20.32**	**14.7**	50.1	**9525**	**20.32**
225	10	0.4	7.5	13.2	2136	**50.25**	7.4	12.4	2464	**50.25**	7.8	15.5	2265	**50.25**
225	10	0.6	32.4	123.2	15537	**34.56**	33.9	90.1	13763	**34.56**	31.5	70.1	15465	**34.56**
225	10	0.8	295 (8)	0.32%	114029	**19.62**	338.7 (7)	12.07%	130079	19.7	344.7 (8)	0.33%	133025	**19.62**
400	4	0.4	7.3	22.3	3345	**55.37**	6.3	15.5	2546	**55.37**	**6.1**	**9.6**	2777	**55.37**
400	4	0.6	**6.7**	**11.9**	4103	**39.04**	7.5	16.7	4044	**39.04**	8.7	25.3	6589	**39.04**
400	4	0.8	9	22.1	5397	**21.03**	11.4	44.9	6263	**21.03**	9.2	19.4	5194	**21.03**
400	7	0.4	**34.4**	144.4	**10464**	**52.05**	48.9	257	15696	**52.05**	37.7	218.2	11164	**52.05**
400	7	0.6	83.4	250.9	27604	**36.12**	**74.5**	209.5	**26944**	**36.12**	78.7	**185.1**	28692	**36.12**
400	7	0.8	**84.4**	187.6	**28762**	**20.19**	98.2	182.5	35369	**20.19**	92.6	206.4	34328	**20.19**
400	10	0.4	68.4	**197.4**	13777	**50.58**	86.7	387.2	17514	**50.58**	91.9	407.6	19024	**50.58**
400	10	0.6	289.4 (9)	0.11%	61886	**34.54**	335 (9)	0.24%	69457	**34.54**	285.7	**563.5**	59428	**34.54**
400	10	0.8	583.5 (**1**)	0.42%	97022	**19.5**	599 (**1**)	0.4%	93171	**19.5**	**577 (1)**	0.43%	97258	19.52

Table 5.10: Results obtained for the three best OWAP basic formulations applied to the Perfect Matching Problem

formulation that includes such valid inequality. All displayed bars represent percentages of mean values computed over 30 random instances with $p = 10$, $|V| = 400$ and $\alpha \in \{0.4, 0.6, 0.8\}$.

From the results displayed in Figure 5.3, we observe first that the gap_{LR} is similar for all formulations but (5.32.1), (5.36.1), (5.37), (5.39.1) and (5.40). As compared with with F_{R2}^{zy}, formulation (5.36.1) improves the values of gap_{LR}, $nodes$ and t, gap. However, (5.32.1), (5.39.1) and (5.40) improve gap_{LR}

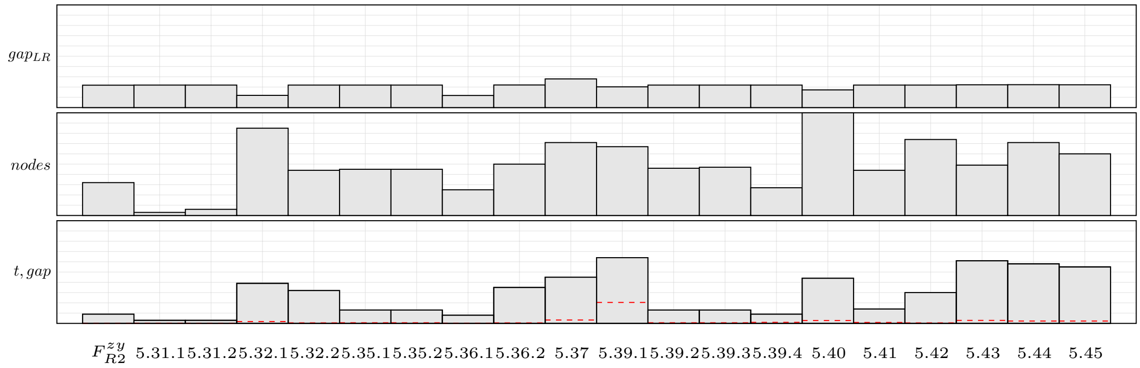


Figure 5.3: Comparative results for the proposed OWAP reinforced formulations applied to the Shortest Path Problem ($p = 10$, $|V| = 400$).

but are not able to improve *nodes* or t, gap. We also note that (5.37) increases gap_{LR} since this gap is computed with a (low quality) best solution found by the solver and the linear relaxation optima at the root node. In addition, formulations (5.31.1), (5.31.2) and (5.39.4) provide promising results in comparison with the values of *nodes* and t, gap.

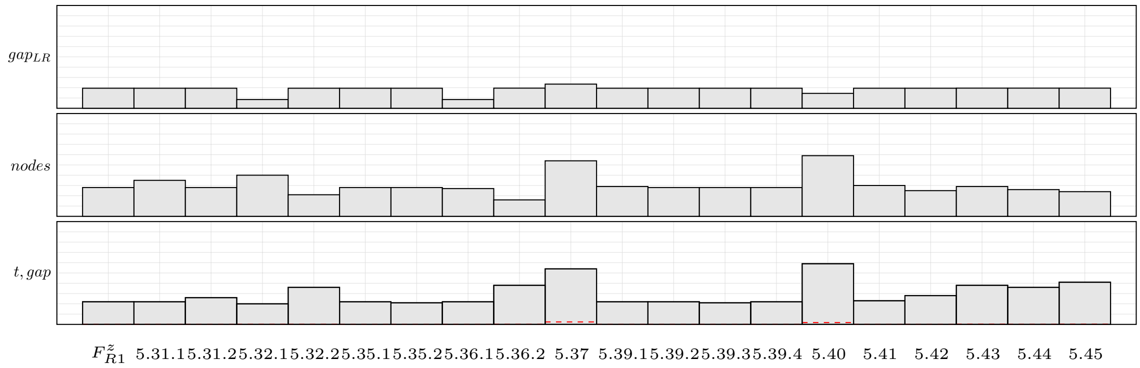


Figure 5.4: Comparative results for the proposed OWAP reinforced formulations applied to the Perfect Matching Problem ($p = 10$, $|V| = 400$).

From the results displayed in Figure 5.4, we observe first that the gap_{LR} is similar for all formulations but (5.32.1), (5.36.1), (5.37) and (5.40). As compared with F^{z}_{R1}, formulations (5.32.1) and (5.36.1), improve gap_{LR} and *nodes* or t, gap. However, (5.40) improves gap_{LR} but is not able to improve *nodes* or t, gap in comparison with the best basic formulation for PMP, namely F^{z}_{R1}. We also note that (5.37) increases gap_{LR} since this gap is computed with a (low quality) best solution found by the solver and the linear relaxation optima at the root node. In addition, formulations (5.35.2), (5.36.2) and (5.39.3) provide promising results in comparison with the values of *nodes* or t, gap.

In summary, we observe the performance of the OWAP formulation depends on its combination with the considered combinatorial object. In particular we conclude, from our computational experience, that for the SPP, it is convenient to apply F^{zy}_{R2} reinforced with (5.31.1) and (5.31.2); although rather similar results can be obtained with F^{z}_{R2}. The conclusion for the PMP is different, because the best basic formulation is now F^{z}_{R1} and the reinforcements (5.32.1). Once more, rather similar results are obtained for F_{R1}^{zy} and F_{R1}^{s}. Therefore, we cannot conclude whether there is a formulation superior to all the others regardless the domain Q to be considered. For this reason it is important to have developed the catalogue of formulations and valid inequalities presented in this chapter. In general, it is advisable to test them depending on the combinatorial object to be considered.

5.7 Conclusions

In this chapter we have presented and revisited different mathematical programming formulations for the OWAP using different sets of decision variables. These formulations reinforced with appropriate constraints have shown to be rather promising for efficiently solving many medium size OWAP instances. However, from the obtained results it is also clear that for solving larger OWAP instances with more objective functions further improvements are needed. Our current research focuses on the design of more sophisticated elimination tests as well as from alternative formulations leading to tighter LP bounds.

Bibliography

Boland, N., Domínguez-Marín, P., Nickel, S. and Puerto, J. (2006) "Exact procedures for solving the discrete ordered median problem". *Computers & Operations Research*, vol. 33 (11): 3270–3300.

Cherkassky, B., Goldberg, A. and Radzik, T. (1996) "Shortest paths algorithms: Theory and experimental evaluation". *Mathematical Programming*, vol. 73: 129–174.

Edmonds, J. (1965) "Paths, trees, and flowers". *Canadian Journal of Mathematics*, vol. 17: 449–467.

Ehrgott, M. (2005) *Multicriteria Optimization*. Springer Berlin Heidelberg.

Fernández, E., Pozo, M. A. P. and Puerto, J. (2013) "A modeling framework for ordered weighted average combinatorial optimization". *arXiv:1306.1426*.

Fernández, E., Puerto, J. and Rodriguez-Chia, A. (2012) "On discrete optimization with ordering". *Annals of Operations Research*. Doi: 10.1007/s10479-011-1044-7.

Galand, L. and Spanjaard, O. (2012) "Exact algorithms for OWA-optimization in multiobjective spanning tree problems". *Computers & Operations Research*, vol. 39: 1540–1554.

Grötschel, M. and Holland, O. (1985) "Solving matching problems with linear programming". *Mathematical Programming*, vol. 33(3): 243–259.

Grzybowski, J., Nickel, S., Pallaschke, D. and Urbanski, R. (2011) "Ordered median functions and symmetries". *Optimization*, vol. 60(7): 801–811.

Hurwicz, L. (1951) "Optimality criteria for decision making under ignorance". *Cowles Commission Discussion Paper, Chicago*, p. 370.

Kasperski, A. and Zielinski, P. (2013) "Combinatorial optimization problems with uncertain costs and the OWA criterion". *arXiv:1305.5339*.

Lei, T. L. and Church, R. L. (2012) "Vector assignment ordered median problem: A unified median problem". *International Regional Science Review*. Doi: 10.1177/0160017612450710.

Lesca, J., Minoux, M. and Perny, P. (2013) "Compact versus noncompact LP formulations for minimizing convex choquet integrals". *Discrete Applied Mathematics*, vol. 161: 184–199.

Marín, A., Nickel, S., Puerto, J. and Velten, S. (2009) "A flexible model and efficient solution strategies for discrete location problems". *Discrete Applied Mathematics*, vol. 157(5): 1128–1145.

Nickel, S. and Puerto, J. (2005) *Location Theory: A Unified Approach.* Springer-Verlag, Heidelberg, Germany.

Ogryczak, W. and Olender, P. (2012) "On MILP models for the OWA optimization". *Journal of Telecommunications and Information Technology*, vol. 2/2012: 5–12.

Ogryczak, W. and Sliwinski, T. (2003) "On solving linear programs with the ordered weighted averaging objective". *European Journal of Operational Research*, vol. 148(1): 80–91.

Ogryczak, W. and Tamir, A. (2003) "Minimizing the sum of the k largest functions in linear time". *Information Processing Letters*, vol. 85 (3): 117–122.

Perny, P. and Spanjaard, O. (2003) "An axiomatic approach to robustness in search problems with multiple scenarios." In *Proceedings of the 19th conference on uncertainty in artificial intelligence*, pp. 469–476.

Puerto, J. and Tamir, A. (2005) "Locating tree-shaped facilities using the ordered median objective". *Mathematical Programming*, vol. 102(2): 313–338.

Ramaswamy, R., Orlin, J. B. and Chakravarti, N. (2005) "Sensitivity analysis for shortest path problems and maximum capacity path problems in undirected graphs". *Mathematical Programming*, vol. 102(2): 355–369.

Schmeidler, D. (1986) "Integral representation without additivity". *Prooceedings of the American Mathematical Society*, vol. 97(2): 255–261.

Weaver, J. R. and Church, R. L. (1985) "A median location model with non-closest facility service". *Transportation Science*, vol. 19: 58–74.

Yager, R. (1988) "On ordered weighted averaging aggregation operators in multicriteria decision making". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18: 183–190.

Yager, R. and Kacprzyk, J. (1997) *The Ordered Weighted Averaging Operators: Theory and Applications.* Kluwer, Norwell, MA.

Chapter 6

Ordered Weighted Average Optimization in multiobjective spanning tree problems

ABSTRACT

Multiobjective Spanning Tree Problems are analyzed in this chapter. In particular, the ordered median objective function is used as an averaging operator to aggregate the vector of objective values of feasible solutions. This leads to the study of the Ordered Weighted Average Spanning Tree Problem, a nonlinear combinatorial optimization problem. Different reformulations as a mixed integer Linear problem are proposed, based on the most relevant Minimum cost Spanning Tree formulations in the literature and on a new one derived from an extended formulation proposed by Kipp Martin. These reformulations are analyzed and several enhancements proposed. Their empirical performance is tested over a set of randomly generated benchmark instances. The results of the computational experiments show that the choice of an appropriate reformulation allows to solve larger instances with more objectives than those previously solved in the literature.

Keywords: Combinatorial Optimization, Multiobjective optimization, Ordered median, Ordered weighted average, Spanning trees.

6.1 Introduction

Optimization problems related to spanning trees, or simply Spanning Tree Problems are among the core problems in combinatorial optimization. On the one hand, the combinatorial object that represents spanning trees has important structural properties. On the other hand, from a practitioner point of view, spanning trees are found in a wide range of applications in many fields (e.g. computer networks design, telecommunications networks, transportation, etc). Furthermore, they often appear as subproblems of other more complex optimization problems.

The most relevant property of trees is their matroid structure. This implies that the basic problem of finding a minimum cost spanning tree, can be solved efficiently (Prim, 1957; Kruskal, 1956). This also implies that formulations with the integrality property can be obtained, which allow to solve the minimum cost Spanning Tree Problem (STP) with linear programming tools. However, these good features can be lost for several reasons. For instance, when the objective function does not preserve Gale optimality, i.e., it is not monotonic on the edges costs (Lawler, 1966; Fernández et al., 2014), as it happens in the Optimum Communication Spanning Tree Problem (Hu, 1974). The reader may refer to Landete and Marín (2014) for a description of alternative objective functions for the STP. An optimization STP also becomes a hard problem when several objectives are considered simultaneously (Ehrgott, 2005). In such cases, no efficient combinatorial algorithm is known so the choice of an appropriate mathematical programming representation of the combinatorial object may become crucial. In this sense, formulations for the STP with good properties can be outperformed by other formulations in the new environment.

From a different point of view, in the multiobjective case, it is widely accepted that the use of order and aggregation functions may yield compromise solutions for the different criteria. The literature includes many works on this area related to combinatorial optimization. Some examples, among many others, include minimax problems (Hansen, 1980; Schrijver, 1983), combining minisum and minimax (Averbakh et al., 1995; Hansen and Labbé, 1988; Hansen et al., 1991; Minoux, 1989; Punnen et al., 1995; Tamir et al., 2002), k-centrum optimization (Garfinkel et al., 2006; Kalcsics et al., 2002; Punnen, 1992; Slater, 1978a,b; Tamir, 2000), lexicographic optimization (Calvete and Mateo, 1998; Croce et al., 1999), k-th best solutions (Lawler, 1972; Martello et al., 1984; Pascoal et al., 2003; Yen, 1971), most uniform solutions (Galil and Schieber, 1998; López de los Mozos et al., 2008), minimum-envy solutions (Espejo et al., 2009), solutions with minimum deviation (Gupta et al., 1990), regret solutions (Averbakh, 2001; Conde, 2004; Puerto and Rodríguez-Chía, 2003), equity measures (Gupta and Punnen, 1988; López de los Mozos et al., 2008; Mesa et al., 2003; Punnen and Aneja, 1997), discrete ordered median location problems (Boland et al., 2006; Marín et al., 2009; Puerto, 2008; Puerto and Tamir, 2005; **?**), ordered weighted average objectives (Fernández et al., 2013, 2014; Galand and Spanjaard, 2012), and covering objectives (Balas and Padberg, 1972; Breuer, 1970; Christofides and Korman, 1974; Kelly, 1944; Lawler, 1966). Among the aggregation functions mentioned above, the Ordered Weighted Average operator (OWA) is particularly relevant, because of its generality, as it includes as particular cases most of the above mentioned operators. This observation has been made explicit in Fernández et al. (2014).

Multiobjective STPs have already been studied by some authors, mostly for the biobjective case (see Hamacher and Ruhe, 1994; Andersen et al., 1996; **?**; Sourd and Spanjaard, 2008; Steiner and Radzik, 2008). In this chapter we address the Multiobjective STP under the perspective of the OWA operator for a general number of objectives. This problem will be referred to as the OWA Spanning Tree Problem (OWASTP). In the OWASTP the optimality of traditional combinatorial algorithms is no longer guaranteed. Furthermore, formulations adapted from good STP formulations lose the integrality property. Thus alternative formulations that originally do not exhibit such good properties, may now outperform them. In Galand and Perny (2007) the OWASTP was addressed using Choquet optimization and Galand and Spanjaard (2012) presented a first ordered median Mixed Integer Linear Programming (MILP) formulation. Our goal in this chapter is to exploit properties of alternative formulations for the OWASTP. As we will see, an appropriate formulation allows us to solve larger instances and with more objectives than those previously solved in the literature (Galand and Spanjaard, 2012), with up to 100 nodes and 10 objective functions. The contributions of this chapter are (1) to provide new formulations for the OWASTP combining appropriate STP and OWAP formulations; (2) to prove a new complexity result according to which the OWASTP is NP-complete even for cactus graphs and two objectives; (3) to establish a theoretical and empirical comparison between the new formulations and previous existing ones; and, (4) to provide reinforcements that together with the new OWASTP formulations are able to outperform previous results in the literature.

The structure of the chapter is the following. In Section 6.2 we formally define the OWASTP and prove our new complexity result. Section 6.3 presents the catalogue of STP formulations that we study for the OWASTP. One such formulation has already been used in Galand and Spanjaard (2012). We will use it as a reference for the alternative formulations that we present. The empirical performance of the resulting OWASTP formulations is analyzed in Section 6.4, where we present extensive numerical results and a comparison with existing ones. Finally, some conclusions are summarized in Section 6.5.

6.2 Problem definition

The Ordered Weighted Average operator is defined over a feasible set $Q \subseteq \mathbb{R}^m$. Let $C \in \mathbb{R}^{p \times m}$ be a given matrix, whose rows, denoted by C^i, are associated with the cost vectors of p objective functions. The index set for the rows of C is denoted by $P = \{1, \ldots, p\}$. Let also $\omega \in \mathbb{R}^{p+}$ denote a vector of non-negative weights. For $x \in Q$, the vector $y = Cx \in \mathbb{R}^p$ is referred to as the outcome vector relative to C. In the following we assume $y = Cx$, with $x \in Q$. For a given y, let σ be a permutation of the indices of $i \in P$ such that $y_{\sigma_1} \geq \ldots \geq y_{\sigma_p}$. Feasible solutions $x \in Q$ are evaluated with an operator defined as $OWA_{(C,\omega)}(x) = \omega' y_\sigma$. The OWA optimization Problem (OWAP) is to find $x \in Q$ of minimum value with respect to the above operator, that is

$$\text{OWAP:} \quad \min_{x \in Q} OWA_{(C,\omega)}(x)$$

The OWA is a very general operator, which has as particular cases well-known objective functions namely the Ordered Median Objective and the Vector Assignment Ordered Median (see Fernández et al., 2014). In addition, the OWA operator allows to model various aggregation functions according to the vector of weights w (see, e.g., Ogryczak and Olender, 2012). Some examples are the minimum, maximum, median, center or k-centrum functions. Therefore, the selection of non/monotonic or non/symmetric w-weights is directly connected with the problem structure and thus with its complexity (Kasperski and Zielinski, 2013).

As defined, the OWA operator is indeed not linear. Moreover, in general, it is not convex either. For the case of monotonic weights, its convexity is known **?** and some elegant linearization of OWA functions have been proposed in the literature (see, e.g., Ogryczak and Sliwinski, 2003; Ogryczak and

Tamir, 2003). Depending on the type of monotonicity, the problems are simpler (with decreasing weights in the case of minimization) or harder (with increasing weights in the case of minimization). In this chapter we focus on the OWASTP with arbitrary weights. Two well-known particular cases of the OWA operator with arbitrary weights are the Hurwicz criterion (Hurwicz, 1951) defined as $\alpha \max_{i \in P} y_i + (1-\alpha) \min_{i \in P} y_i$ and the k-trimmed mean defined as $\sum_{i=k-1}^{p-k} (p-2k)^{-1} y_i$. These criteria are of special interest for being non-monotonic and non-convex (Grzybowski et al., 2011, Puerto and Tamir, 2005) and have already been considered when analyzing the behavior of OWA operators in multiobjective optimization (see, e.g., Galand and Spanjaard, 2012).

The OWASTP is defined as follows. Let $G = (V, E)$ be an undirected connected graph with set of nodes V, $|V| = n$, and set of edges E, $|E| = m$. In the following we assume that G contains at least one cycle, that is $m > n - 1$, as otherwise the problem becomes trivial. A spanning tree of G is a subgraph $T = (V, E')$ where $E' \subset E$ is a minimal set of edges connecting the set of nodes V. Let $\mathcal{T}$ denote the set of spanning trees defined on G. Then, the OWASTP can be defined as

$$\text{OWASTP:} \quad \min_{x \in \mathcal{T}} OWA_{(C,\omega)}(x).$$

Example 6.1-

Consider the graph $G = (N, E)$ depicted in Figure 6.1-(a) and the 3-cost vectors on E, whose values are represented next to each edge. The optimal solution to the OWASTP with $\omega' = (0.4, 0, 0.6)$ is depicted in Figure 6.1-(b) and has a value of 8.8. When the weights are $\omega' = (0.8, 0, 0.2)$ the optimal OWASTP value is 10.4, corresponding to the tree depicted in Figure 6.1-(c).

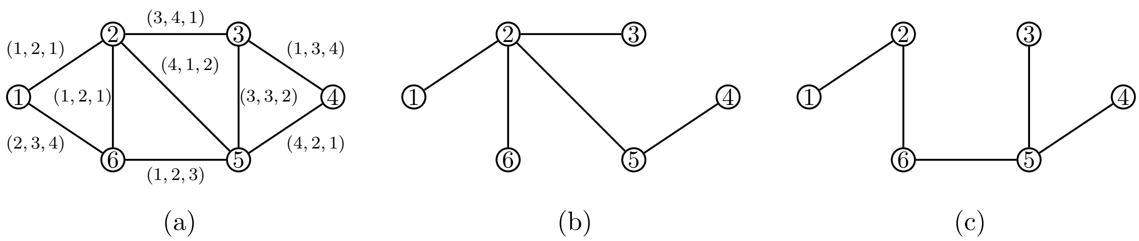


Figure 6.1: Graph with edge costs (a) and OWASTP solutions for $\omega' = (0.4, 0, 0.6)$ (b) and $\omega' = (0.8, 0, 0.2)$ (c).

□

OWASTP is known to be NP-hard on general graphs (Hamacher and Ruhe, 1994; Yu, 1998). We can give, however, a stronger complexity result since as we prove below the OWASTP is NP-complete even for two objective functions ($p = 2$) and on cactus graphs. The reader may note that *cactus* graphs are considered as *almost trees* and classified in the lowest level difficulty class of graphs, just after the acyclic ones (see, e.g., graphclasses.com).

The OWASTP problem in decisional form is: Given a graph $G = (V, E)$ with weights (c_e^1, c_e^2) assigned to each edge $e \in E$. Let $f_1(T)$ and $f_2(T)$ be the weights of a spanning tree T computed w.r.t. c_e^1 and

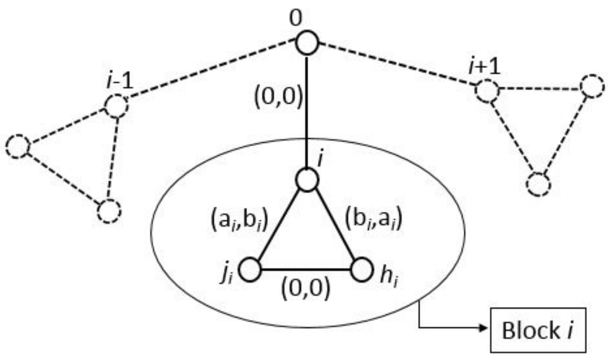


Figure 6.2: The Cactus graph used in proof of the NP-completeness claim.

c_e^2, for all $e \in V(T)$, respectively. Is there a spanning tree T of G and weights $(w_1, w_2) \geq 0$ such that for an ordering σ of $f_1(T)$ and $f_2(T)$ one has $w_1 f_{\sigma_1}(T) + w_2 f_{\sigma_2}(T) \leq K$?

Claim Problem OWASTP is NP-complete on cactus graphs and $p = 2$.

The reduction is from Partition with Disjoint Pairs (PDP) which is the following problem: Given n pairs of integers (a_i, b_i), $i = 1, \ldots, n$, is there a subset $S \subset [1, \ldots, n]$ (a bi-partition) of the set of indices such that: $\sum_{i \in S} a_i + \sum_{i \notin S} b_i = \frac{Q}{2}$, where $\sum_{i=1}^{n} (a_i + b_i) = Q$?

Proof. Given an instance of PDP, construct the (very simple) cactus graph in Figure 6.2. Set $w_1 = 1$, $w_2 = 1 - \epsilon$, for $\epsilon > 0$, and $K \leq Q(1 - \frac{\epsilon}{2})$.

To each block i assign to edge (i, j_i) weights (a_i, b_i); assign weights (b_i, a_i) to edge (i, h_i). To all other edges of G assign weights $(0, 0)$. Given a solution of PDP, we can construct a solution for OWASTP as follows: if $i \in S$ the corresponding edge in the spanning tree T in block i is (i, j_i), otherwise add to T (i, h_i). Then, since T must be a spanning tree of G, we must add edges (j_i, h_i) and all the edges $(i, 0)$, $i = 1, \ldots, n$. The weight of T w.r.t. the first component of the edge weights (i.e., $f_1(T)$) is $f_1(T) = \sum_{i \in S} a_i + \sum_{i \notin S} b_i = \frac{Q}{2}$. Thus, $f_2(T) = \sum_{i \notin S} a_i + \sum_{i \in S} b_i = \frac{Q}{2}$. Hence,

$$f_{\sigma_1}(T) + (1 - \epsilon) f_{\sigma_2}(T) = Q(1 - \frac{\epsilon}{2}) = K.$$

Conversely, if we have a solution of OWASTP, there must exist a subset S of the n blocks for which edges (a_i, b_i) are added to T, $i \in S$; while for the other blocks (i.e., $i \notin S$) edges (b_i, a_i) belong to T. In fact, since $\sum_{i=1}^{n} (a_i + b_i) = Q$, any other different assignment of edges in (at least one) block i, will produce a solution for OWASTP such that $f_{\sigma_1}(T) > \frac{Q}{2}$ and $f_{\sigma_2}(T) < \frac{Q}{2}$. Suppose that $f_{\sigma_1}(T) = \frac{Q}{2} + \Delta$ and $f_{\sigma_2}(T) = \frac{Q}{2} - \Delta$, $\Delta > 0$. Then, computing the objective function of OWASTP we have:

$$f_{\sigma_1}(T) + (1 - \epsilon) f_{\sigma_2}(T) = \frac{Q}{2} + \Delta + (1 - \epsilon)(\frac{Q}{2} - \Delta) = Q(1 - \frac{\epsilon}{2}) + \epsilon\Delta > K.$$

Since OWASTP (clearly) belongs to NP, then OWASTP is NP-complete.

6.3 OWASTP formulations

In this section we present several formulations for the OWASTP. All of them are MILP formulations, which integrate a mixed integer linear programming STP formulation within a generic mathematical programming formulation for an OWA combinatorial problem (Fernández et al., 2014). We start with the catalogue of STP formulations and then we give the mathematical programming formulations for OWA combinatorial optimization problems that we have used.

6.3.1 Mixed Integer Linear Programming formulations for the STP

Many alternative MILP formulations have been proposed for the STP. For an overview of the possible alternatives and the properties in each case, the interested reader is addressed to the excellent book chapter by Magnanti and Wolsey (1995) where many of them are presented and compared.

It is well-known that STP formulations exist with the integrality property. Unfortunately, when they are embedded within the OWAP framework the integrality property is lost, so explicit integrality conditions are needed. Alternative STP formulations without such property may now be superior. This explains why some of the formulations we have used lack the integrality property. The criterion that has guided the selection of the formulations is either their good theoretical properties or some characteristic that seemed useful as, for instance, a small number of variables or constraints.

We start with two well-known models, the first one derived from the matroid polyhedron (Edmonds, 1970, 1971) and the second one proposed by Martin (1991), both of which having the integrality property. Then we present three existing formulations without the integrality property, based, respectively, on cutset inequalities, flow balance equations and Miller-Tuker-Zemlin inequalities (Miller et al., 1960). We present another STP formulation based on a relaxation of the formulation proposed by Martin (1991), which uses considerably fewer variables.

All formulations use design variables x to represent the edges of the spanning trees. Let $x_e, e \in E$ be a binary variable equal to 1 if edge $e = (u, v)$ is in the spanning tree, and zero otherwise. Some formulations use additional variables related to the arcs of the directed network, $D = (V, A)$ with the same node set as the original undirected G and set of arcs A, containing two arcs associated with each edge of E, i.e., $A = \{(u, v), (v, u) \mid (u, v) \in E\}$.

Throughout we will use the following standard notation. Given a subset of nodes $S \subset V$, $E(S)$ and $A(S)$ respectively denote the subsets of edges of E and arcs of A with both end-nodes in S, i.e., $E(S) = \{e = (u, v) \in E : u, v \in S\}$ and $A(S) = \{(u, v) \in A : u, v \in S\}$. The *cut-set associated with* $S \subset V$, $\delta(S) = \{e = (u, v) \in E \mid (u \in S, v \in V \setminus S) \text{ or } (v \in S, u \in V \setminus S)\}$, contains all edges with one node in S and the other node outside S. When working on the directed network D, for $S \subset V$, we let $\delta^+(S) = \{(u, v) \in A \mid u \in S, v \in V \setminus S\}$ denote the cutset directed out of S and $\delta^-(S) = \{(u, v) \in A \mid u \in V \setminus S, v \in S\}$ the cutset directed into S. Directed cuts will also be referred to as *dicuts*.

Next we focus on the domains that characterize feasible solutions in each case.

The domain in the subtour elimination formulation is:

$$\mathcal{T}^{sub}: \sum_{e \in E} x_e = n - 1 \tag{6.1a}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \qquad \emptyset \neq S \subset V \tag{6.1b}$$

$$x_e \geq 0 \qquad e \in E \tag{6.1c}$$

The cardinality constraint (6.1a) imposes that exactly $n-1$ edges are chosen. Constraints (6.1b) ensure that the solution contains no cycle. The number of such constraints is exponential on the number of nodes. However, they can be separated in polynomial time by solving a series of minimum (s,t)-cut problems. An effective algorithm can be implemented using a Gomory-Hu cut tree (Hu, 1974).

It is well-known that all the extreme points in the above domain are integer and that formulation $\mathcal{T}^{sub}$ is stronger than the formulation where inequalities (6.1b) are replaced by the cut-set constraints $\sum_{e \in \delta(S)} x_e \geq 1$, that we denote $\mathcal{T}^{cut}$, which may have fractional extreme points (Magnanti and Wolsey, 1995).

The extended formulation of Martin (1991) models an arborescence rooted at each node $k \in V$, in which arcs follow the direction from the leaves to the root. The arcs of such arborescences are then related to the design x variables. For $k \in V, (u,v) \in E$, let q_{kuv} and q_{kvu} be decision variables that, respectively, indicate whether or not arcs (u,v) and $(v,u) \in A$ belong to the arborescence rooted at k. The domain of the K. Martin (KM) formulation is the following:

$$\mathcal{T}^{km}: \sum_{e \in E} x_e = n - 1 \tag{6.2a}$$

$$q_{kuv} + q_{kvu} = x_{uv} \qquad k \in V, (u,v) \in E \tag{6.2b}$$

$$\sum_{(u,v) \in \delta^+(u)} q_{kuv} \leq 1 \qquad k \in V, u \in V : u \neq k \tag{6.2c}$$

$$\sum_{(k,v) \in \delta^+(k)} q_{kkv} \leq 0 \qquad k \in V \tag{6.2d}$$

$$x_e \geq 0 \qquad e \in E \tag{6.2e}$$

$$q_{kuv} \geq 0 \qquad k \in V, (u,v) \in A \tag{6.2f}$$

Constraint (6.2a) ensures that the tree has $n-1$ edges. On the other hand, constraints (6.2b) indicate that the arcs that are used in the arborescences are precisely the ones associated with the $n-1$ selected undirected edges. In other words, the underlying undirected graph supporting all the arborescences is exactly the same, so all the arborescences use exactly $n-1$ arcs, and the only differences among arborescences are the directions of the arcs, but not the edges on the undirected graph that are used. For each arborescence, (6.2c) impose that no more than one arc leaves any node different from the root k, while (6.2d) forbids any arc leaving the root node k. Hence, these constraints imply that for each arborescence, the component containing the root node does not contain any cycle. Since each node is the root of one arborescence, (6.2b) guarantee that the selected undirected edges contain no cycle and, by (6.2a), the solutions define spanning trees.

While formulation $\mathcal{T}^{km}$ has the integrality property, it has an $O(n^3)$ number of both q variables and constraints (6.2b). As the size of the graph increases, this number can be prohibitive. When the integrality property is lost because of the addition of new constraints, the computational burden for solving a formulation with such a large number of variables and constraints may become too high.

The Miller-Tucker-Zemlin (MTZ) inequalities are an alternative to the exponential size family of constraints in (6.1b), to guarantee the connectivity of the solutions and thus prevent cycles. These constraints were initially proposed by Miller et al. (1960) in the context of the Traveling Salesman Problem. They have been adapted to other problems and reinforced by different authors (see, e.g. Laporte, 1992, Landete and Marín, 2014). In particular, they have been used by Gouveia (1995) for the Hop-Constrained Spanning Tree Problem, which is a generalization of the STP in which the paths starting at a specified root node r are restricted to have no more than p edges. The MTZ formulation for the STP builds an arborescence rooted at a specified node $r \in V$, in which arcs follow the direction from the root to the leaves. It uses binary variables to represent the arcs of the arborescence. Each edge $(u, v) \in E$, is associated with a pair of binary variables, y_{uv} and y_{vu}, which take the value 1 if and only if arcs (u, v) and $(v, u) \in A$ belong to the arborescence, respectively. In addition, it uses continuous variables l_u, denoting the position that node u occupies in the arborescence with respect to r. Since, in principle, there is no pre-specified root node, below r denotes any arbitrarily selected node. The domain of this formulation is given by the following set of constraints:

$$\mathcal{T}^{mtz}: \sum_{e\in E} x_e = n-1 \tag{6.3a}$$

$$\sum_{(v,u)\in\delta^-(u)} y_{vu} = 1 \qquad u \in V \setminus \{r\} \tag{6.3b}$$

$$y_{uv} + y_{vu} = x_{uv} \qquad (u, v) \in E \tag{6.3c}$$

$$l_v \geq l_u + 1 - n(1 - y_{uv}) \qquad (u, v) \in A \tag{6.3d}$$

$$l_u = 1 \qquad u = r \tag{6.3e}$$

$$2 \leq l_u \leq n \qquad u \in V \setminus \{r\} \tag{6.3f}$$

$$y_{uv} \in \{0, 1\} \qquad (u, v) \in A \tag{6.3g}$$

$$x_e \in \{0, 1\} \qquad e \in E \tag{6.3h}$$

Constraint (6.3a) ensures that the tree has $n-1$ edges. Equations (6.3b) impose that each node apart from the root is reached by one single arc, while (6.3c) guarantee that an edge is selected if any of its two arcs is selected. Constraints (6.3d) state that if an arc (u, v) is selected the position in the tree of v is higher than the position of u. Finally, (6.3e) and (6.3f) assign appropriate bounds to variables l_u, to ensure that the relative position of the root node in the tree is 1 and that the position of any other node is greater than or equal to 2 and does not exceed the number of nodes.

The flow-based STP formulation we present below (see Magnanti and Wolsey, 1995) is based on the formulation of Gavish (1983) for the capacitated minimal directed tree problem, and was used by Galand and Spanjaard (2012) for the OWASTP. In addition to the binary design variables x, the formulation uses continuous flow variables φ defined on the arcs of the directed network $D = (V, A)$. There is a single source node, which is an arbitrarily selected node $r \in V$, with inflow $n-1$. All other nodes have a demand of one unit. For each $(u, v) \in A$ the decision variable φ_{uv} represents the amount of flow through arc (u, v). Then the domain of the flow formulation for the STP is:

$$\mathcal{T}^{flow}: \sum_{e \in E} x_e = n - 1 \tag{6.4a}$$

$$\sum_{(r,v) \in \delta^+(r)} \varphi_{rv} - \sum_{(u,r) \in \delta^-(r)} \varphi_{ur} = n - 1 \tag{6.4b}$$

$$\sum_{(u,v) \in \delta^+(u)} \varphi_{uv} - \sum_{(v,u) \in \delta^-(u)} \varphi_{vu} = -1 \qquad u \in V \setminus \{r\} \tag{6.4c}$$

$$\varphi_{uv} \leq (n-1)x_{uv} \qquad (u, v) \in E \tag{6.4d}$$

$$\varphi_{vu} \leq (n-1)x_{uv} \qquad (u, v) \in E \tag{6.4e}$$

$$\varphi_{uv} \geq 0 \qquad (u, v) \in A \tag{6.4f}$$

$$x_e \in \{0, 1\} \qquad e \in E \tag{6.4g}$$

Again, constraint (6.4a) ensures that exactly $n-1$ edges are selected. The block of constraints (6.4b)–(6.4c) guarantees that $n-1$ units of flow leave the source node r and that at least one unit of flow arrives to every other node. The main role of these constraints is to guarantee that the graph induced by the arcs through which the flow circulates is connected and all nodes are "covered". Constraints (6.4d)–(6.4e) extend these two properties to the graph induced by the x variables, by imposing that all the edges used for sending flow in some direction are activated.

Concerning domain $\mathcal{T}^{flow}$ note that, because of the flow constraints (6.4b)–(6.4c), the removal of constraint (6.4a) would not change the set of optimal solutions (as opposed to the case of the maximum STP). However, constraint (6.4a) reinforces considerably the linear relaxation of formulation $\mathcal{T}^{flow}$, so it is kept in the formulation. Another improvement consists of replacing (6.4d) and (6.4e) by the tighter set of constraints:

$$\varphi_{uv} + \varphi_{vu} \leq (n-1)x_{uv} \qquad (u,v) \in E : u = r \vee v = r \qquad (6.4d')$$

$$\varphi_{uv} + \varphi_{vu} \leq (n-2)x_{uv} \qquad (u,v) \in E : u \neq r \wedge v \neq r \qquad (6.4e')$$

An alternative formulation for the STP

Below we present an alternative formulation for STPs, which inherits some of the ideas behind the $\mathcal{T}^{km}$ formulation without requiring $O(n^3)$ variables. In particular, instead of building one arborescence for each node, we arbitrarily set one single root node $r \in V$ and build one single arborescence rooted at r. The arcs of such an arborescence are determined by the subset of variables q_{ruv}, $(u,v) \in A$. Since r is fixed, in the following we remove the first index and simply denote these variables by q_{uv}, $(u,v) \in A$. Indeed, equality (6.2a), plus the subset of constraints (6.2b), (6.2c) and (6.2d) associated with $k = r$ defines a relaxation to formulation $\mathcal{T}^{km}$, which uses $O(n^2)$ variables. Unfortunately, such relaxation is not valid for STPs, as it may produce solutions which are not associated with connected sets of arcs. Luckily, this weakness can be easily overcome by including the following dicut inequalities:

$$\sum_{(u,v)\in\delta^+(S)} q_{uv} \geq 1, \qquad S \subseteq V \setminus \{r\},$$

which guarantee the connectivity of the obtained solutions (at least one arc will *exit* from any subset of nodes S not containing the root node) and thus, the validity of the formulation. The formulation is then as follows:

$$\mathcal{T}^{km2}: \sum_{(u,v)\in E} x_{uv} = n-1 \qquad (6.6a)$$

$$q_{uv} + q_{vu} = x_{uv} \qquad (u,v) \in E \qquad (6.6b)$$

$$\sum_{(u,v)\in\delta^+(u)} q_{uv} \leq 1 \qquad u \in V \setminus \{r\} \qquad (6.6c)$$

$$\sum_{(r,v)\in\delta^+(r)} q_{rv} \leq 0 \qquad (6.6d)$$

$$\sum_{(u,v)\in\delta^+(S)} q_{uv} \geq 1 \qquad \emptyset \neq S \subset V \setminus \{r\} \qquad (6.6e)$$

$$x_{uv} \geq 0 \qquad (u,v) \in E \qquad (6.6f)$$

$$q_{uv} \geq 0 \qquad u,v \in V \qquad (6.6g)$$

Remark 6.1-

(a) The only difference between formulations $\mathcal{T}^{mtz}$ and $\mathcal{T}^{km2}$ is the way in which subtours are prevented. The former uses the Miller-Tucker-Zemlin inequalities, which are known to be weaker than cut-type constraints used in the latter. This indicates that formulation $\mathcal{T}^{mtz}$ is weaker than $\mathcal{T}^{km2}$. Below we provide a stronger evidence of the superiority of $\mathcal{T}^{km2}$ over $\mathcal{T}^{mtz}$, as we will see that $\mathcal{T}^{km2}$ has the integrality property, even if some redundancies are eliminated.

(b) For any $u \in V \setminus \{r\}$ the constraint (6.6e) corresponding to the set $S = \{u\}$ reduces to $\sum_{(u,v)\in\delta^+(u)} q_{uv} \geq 1$. Together with constraints (6.6c) this implies that $\sum_{(u,v)\in\delta^+(u)} q_{uv} = 1$ for all $u \in V \setminus \{r\}$. Observe, however, that the new set of constraints (6.6e) together with (6.6a) and (6.6b) already imply that $\sum_{(u,v)\in\delta^+(u)} q_{uv} = 1$ for all $u \in V \setminus \{r\}$. To see this, note first that if we add all the constraints (6.6e) associated with singletons $S = \{u\}$ with $u \in V \setminus \{r\}$ we get

$$\sum_{u\in V\setminus\{r\}} \sum_{(u,v)\in\delta^+(u)} q_{uv} = \sum_{(u,r)\in\delta^-(r)} q_{ur} + \sum_{(u,v)\in A(V\setminus\{r\})} q_{uv} \geq n-1.$$

Thus, we have,

$$n-1 \leq \sum_{(u,r)\in\delta^-(r)} q_{ur} + \sum_{(u,v)\in A(V\setminus\{r\})} q_{uv} \leq$$

$$\sum_{(r,v)\in\delta^+(r)} q_{rv} + \sum_{(u,r)\in\delta^-(r)} q_{ur} + \sum_{(u,v)\in A(V\setminus\{r\})} q_{uv} = \sum_{(u,v)\in A} q_{uv} = \sum_{(u,v)\in E} x_{uv} = n-1,$$

where the last two equalities follow from constraints (6.6b) and (6.6a), respectively.

Hence, we can conclude that $\sum_{(u,v)\in\delta^+(r)} q_{uv} = 0$ and $\sum_{(u,v)\in\delta^+(u)} q_{uv} = 1$ for all $u \in V \setminus \{r\}$, since otherwise we would reach a contradiction.

The above remark indicates that the dicut constraints (6.6e) make the sets of constraints (6.6c) and (6.6d) unnecessary. Hence, the STP formulation which emanates from the above discussion is:

$$\mathcal{T}^{dc}: \sum_{(u,v)\in E} x_{uv} = n-1 \tag{6.7a}$$

$$q_{uv} + q_{vu} = x_{uv} \qquad (u,v)\in E \tag{6.7b}$$

$$\sum_{(u,v)\in\delta^+(S)} q_{uv} \geq 1 \qquad \emptyset \neq S \subset V\setminus\{r\} \tag{6.7c}$$

$$x_{uv} \geq 0 \qquad (u,v)\in E \tag{6.7d}$$

$$q_{uv} \geq 0 \qquad u,v\in V \tag{6.7e}$$

The reader may observe that formulation $\mathcal{T}^{dc}$ can be readily transformed into the *directed cut* formulation of Magnanti and Wolsey (1995) by just changing the directions of the arcs of the arborescence and, thus, directing the arcs from the root r to the leaves, instead of from the leaves to the root. Since the directed cut formulation of Magnanti and Wolsey (1995) has the integrality property, so does formulation $\mathcal{T}^{dc}$. In its turn, this implies the integrality of the domain of $\mathcal{T}^{km2}$.

The number of dicut constraints (6.7c) is exponential on $|V|$. Nevertheless, they can be incorporated into the formulation only if needed via an efficient separation oracle, as they can be separated in polynomial time by finding the Gomory-Hu cut tree (Hu, 1974).

Comparison of formulations

Let $P(\mathcal{T}^{(\cdot)})$ denote the polyhedron associated with the linear programming relaxation of formulation $\mathcal{T}^{(\cdot)}$. Except for formulation $\mathcal{T}^{sub}$, all other formulations above are extended formulations, in the sense that, besides the design x variables, additional sets of variables are used. For comparing all the formulations in the same space we project the polyhedra associated with the extended formulations onto the space of the x variables, and denote by $P_x(\mathcal{T}^{(\cdot)})$ the projected polyhedron associated with formulation $\mathcal{T}^{(\cdot)}$.

Several of the formulations described above have the integrality property, namely formulations $\mathcal{T}^{sub}$, $\mathcal{T}^{km}$ and $\mathcal{T}^{km2}$. This means that $P_x(\mathcal{T}^{sub}) = P_x(\mathcal{T}^{km}) = P_x(\mathcal{T}^{km2})$. In its turn, each of these formulations is tighter than any of the formulations without integrality property. That is, $P_x(\mathcal{T}^{km2}) \subset P_x(\mathcal{T}^{mtz})$ and $P_x(\mathcal{T}^{km2}) \subset P_x(\mathcal{T}^{flow})$. Below we compare $P_x(\mathcal{T}^{mtz})$ and $P_x(\mathcal{T}^{flow})$, as we have not seen such comparison in the literature.

The example of Figure 6.3 illustrates that $P_x(\mathcal{T}^{flow}) \nsubseteq P_x(\mathcal{T}^{mtz})$. The components of a x vector such that $\sum_{e\in E} x_e = n-1$ are given next to each edge. Taking $r = 5$ as the root node, the flow $\varphi_{53} = \varphi_{54} = 2$, $\varphi_{31} = \varphi_{42} = 1$, together with x, define a feasible solution to formulation $P_x(\mathcal{T}^{flow})$. However, there is no feasible y vector that together with the depicted x vector satisfies constraints (6.3b) and (6.3c).

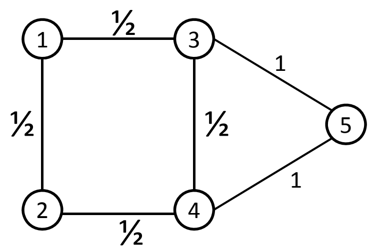


Figure 6.3: Fractional x solution with $\sum_{e\in E} x_e = n-1$.

On the other hand, the example depicted in Figure 6.4 shows that $P_x(\mathcal{T}^{mtz}) \nsubseteq P_x(\mathcal{T}^{flow})$, i.e. the two formulations are not related in that there exist feasible solutions to $P_x(\mathcal{T}^{flow})$ that do not give rise to feasible solutions to $P_x(\mathcal{T}^{mtz})$ and the other way around.

Consider a complete graph with $n = 5$ nodes and cost matrix:

$$C = \begin{pmatrix} 0 & 31 & 19 & 33 & 67 \\ 31 & 0 & 57 & 40 & 38 \\ 19 & 57 & 0 & 2 & 18 \\ 33 & 40 & 2 & 0 & 13 \\ 67 & 38 & 18 & 13 & 0 \end{pmatrix}$$

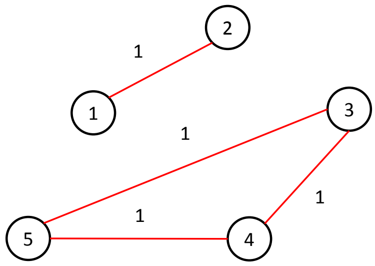


Figure 6.4: Solution x of $\mathcal{T}^{mtz}$ formulation in the complete graph ($n = 5$) of the above example.

The optimal solution to the linear relaxation of $\mathcal{T}^{mtz}$ is given by: $x_{12} = 1$, $x_{34} = 1$, $x_{35} = 1$, $x_{45} = 1$; $y_{12} = 1$, $y_{34} = 0.5$, $y_{35} = 0.5$, $y_{43} = 0.5$, $y_{45} = 0.5$, $y_{53} = 0.5$, $y_{54} = 0.5$; and $\ell_1 = 1$, $\ell_2 = 2$, $\ell_3 = 2$, $\ell_4 = 2$, $\ell_5 = 3.5$.

It is clear that the above solution to $\mathcal{T}^{mtz}$ does not induce a feasible solution to $\mathcal{T}^{flow}$ since the vector x does not produce a connected solution in the graph. Thus we have the following result:

Corollary 6.1-

$$P_x(\mathcal{T}^{sub}) = P_x(\mathcal{T}^{km}) = P_x(\mathcal{T}^{km2}) \subseteq \begin{cases} P_x(\mathcal{T}^{mtz}) \\ \neq \\ P_x(\mathcal{T}^{flow}) \end{cases}$$

6.3.2 Mixed Integer Linear Programming formulations for the OWAP

This section presents the OWA formulation that we use for the OWASTP. The choice is based on our preliminary experiments for the STP and on previous results of Fernández et al. (2014), who show that this formulation outperforms other alternatives when the embedded combinatorial object is the shortest path or the perfect matching problem.

Consider the following binary variables that define the specific positions in the ordering of the sorted cost function values:

$$z_{ij} = \begin{cases} 1 & \text{if cost function } i \text{ occupies position } j, \\ 0 & \text{otherwise} \end{cases}$$

For each $j \in P$, let also θ_j be a variable representing the value of the objective function sorted at position j. Then, the OWAP can be formulated as:

$$
\begin{aligned}
F^{\theta}: \quad & V = \min \sum_{j \in P} \omega_j \theta_j && && (6.8a)\\
& s.t. \sum_{i \in P} z_{ij} = 1 && j \in P && (6.8b)\\
& \sum_{j \in P} z_{ij} = 1 && i \in P && (6.8c)\\
& C^i x \leq \theta_j + M(1 - \sum_{k \geq j} z_{ik}) && i, j \in P && (6.8d)\\
& \theta_j \geq \theta_{j+1} && j \in P : j < p && (6.8e)\\
& x \in \mathcal{T} && && (6.8f)\\
& \theta_j \geq 0 && j \in P && (6.8g)\\
& z \in \{0,1\}^{p \times p} && && (6.8h)
\end{aligned}
$$

The objective function (6.8a) minimizes the weighted average of sorted objective function values, provided that θ_j, $j \in P$, are enforced to take on the appropriate values. Constraints (6.8b)–(6.8c) define a permutation of the cost functions, by placing one single cost function at each position and each cost function at one single position of the sequence. Constraints (6.8d) relate cost function values with the values placed in the sorted sequence. Constraints (6.8e) are optimality cuts which help the resolution of F^{θ}, as explained in Fernández et al. (2014).

For comparison purposes in our computational experiments, below we present the formulation used by Galand and Spanjaard (2012) for the OWASTP. This formulation uses the above z binary variables plus an additional set of continuous variables $y = (y_{ij})_{i,j \in P} \in \mathbb{R}^{p \times p}$, where y_{ij} denotes the value of cost function i if it occupies the j-th position in the ordering. The formulation is as follows:

$$F^{GS}: \quad V = \min \sum_{j \in P} \omega_j \sum_{i \in P} y_{ij} \tag{6.9a}$$

$$s.t. \quad \sum_{i \in P} z_{ij} = 1 \qquad j \in P \tag{6.9b}$$

$$\sum_{j \in P} z_{ij} = 1 \qquad i \in P \tag{6.9c}$$

$$\sum_{i \in P} y_{ij} \geq \sum_{i \in P} y_{ij+1} \qquad j \in P : j < p \tag{6.9d}$$

$$y_{ij} \leq M z_{ij} \qquad i, j \in P \tag{6.9e}$$

$$\sum_{j \in P} y_{ij} = C^i x \qquad i \in P \tag{6.9f}$$

$$x \in \mathcal{T} \tag{6.9g}$$

$$y_{ij} \geq 0 \qquad i, j \in P \tag{6.9h}$$

$$z \in \{0,1\}^{p \times p} \tag{6.9i}$$

The objective function (6.9a) minimizes the weighted average of sorted objective function values. Constraints (6.9b)–(6.9c) are a copy of constraints (6.8b)–(6.8c) respectively, and thus define a cost function permutation. Constraints (6.9d) impose that the sorted values are ordered non-increasingly. Constraints (6.9e) relate cost function values with the values placed in the sorted sequence. Constraints (6.9f) ensure that one of the y_{ij} variables gives precisely the value of the objective function i.

Note that the relationship between θ in formulation F^θ and the y variables in F^{GS} is:

$$\theta_j = \sum_{i \in P} y_{ij} \qquad j \in P : j > 1. \tag{6.10}$$

Next, we prove two results concerning formulations F^θ and F^{GS}. Let us denote by Ω^θ and Ω^{GS} the domains defined by their respective sets of constraints. We first prove that F^θ and F^{GS} have the same set of optimal solutions although $\Omega^{GS} \subset \Omega^\theta$. This property no longer holds for the respective relaxations, where everything remains unchanged except for the z variables, which are allowed to take continuous values, i.e. $z_{i,j} \geq 0$, $i, j \in P$. In particular, we will see that $\Omega^{GS}_{LR} \subset \Omega^\theta_{LR}$, where Ω^θ_{LR} and Ω^{GS}_{LR} denote the respective continuous relaxed domains. Moreover, in general, the sets of optimal solutions of the linear relaxations for the objective functions (6.8a) and (6.9a) do not coincide.

Property 6.1- Every optimal solution to F^{GS} is also optimal to F^θ and conversely.

`Proof.`
Given the relationship (6.10) between θ and y variables, in Ω^θ we can substitute Constraints (6.8d) by $C^i x \leq \sum_{i \in P} y_{ij} + M(1 - \sum_{k \geq j} z_{ik})$, $i, j \in P$.

- We prove first that $\Omega^{GS} \subseteq \Omega^{\theta}$, that is, every solution $(x, z, y) \in \Omega^{GS}$ (not necessarily optimal) is such that $(x, z, y) \in \Omega^{\theta}$. Observe that it suffices to prove that every $(x, z, y) \in \Omega^{GS}$, with y and θ related by (6.10), satisfies

$$C^i x \leq \sum_{i \in P} y_{ij} + M(1 - \sum_{k \geq j} z_{ik}) \qquad i, j \in P. \qquad \text{(6.9d')}$$

  Let $\hat{x}$ be a feasible solution in $\mathcal{T}$ and $\hat{z}$ a permutation that sorts the cost function values of x. Then, for fixed $\hat{x}$ and $\hat{z}$ values there is a unique $\hat{y}$ since, according to (6.9e)–(6.9f) there is at most one $j \in P$ such that $y_{ij} \neq 0$ for each $i \in P$. It follows that such $(\hat{x}, \hat{z}, \hat{y})$ verifies (6.9d').

- Next we prove that every optimal solution of F^{θ}, satisfies that $(x, z, y) \in \Omega^{GS}$, after performing the change of variable given by (6.10). For this, it is sufficient to prove that every optimal solution $(x, z, y) \in \Omega^{\theta}$ verifies (6.9e) and (6.9f). Let $\hat{x}$ be a feasible solution in $\mathcal{T}$. Then, there exists a permutation $\hat{z}$ that sorts the cost function values of $\hat{x}$ in non increasing order. Now, we give values to the $\hat{\theta}$ variables according to this ordered sequence, and we determine the $\hat{y}$ values by means of $\hat{y}_{ij} = \hat{\theta}_j \hat{z}_{ij}$ $i, j \in P$. From here, it follows that $(\hat{x}, \hat{z}, \hat{y})$ verifies (6.9d)–(6.9f). In addition, we note that, in general, for fixed $\hat{x}$ and $\hat{z}$, the polyhedron given by (6.8d)–(6.8e) is unbounded and thus $\Omega^{\theta} \not\subseteq \Omega^{GS}$.

□

Property 6.2- $\Omega^{GS}_{LR} \subset \Omega^{\theta}_{LR}$.

`Proof.`
First, we observe that $\Omega^{\theta}_{LR} \not\subseteq \Omega^{GS}_{LR}$ since, otherwise, the optimal solution of F^{θ}_{LR} for the graph in Example 6.1 (with value 8.6 when $\omega = (0.8, 0, 0.2)$) could not have a smaller value than the optimal solution of F^{GS}_{LR} (with value 9.4).
Next, we prove that every feasible solution $(x, z, y) \in \Omega^{GS}_{LR}$ is such that $(x, z, y) \in \Omega^{\theta}_{LR}$, once the change of variable given by (6.10) is done.
Indeed, we have to prove that any $(x, z, y) \in \Omega^{GS}_{LR}$ verifies

$$C^i x \leq \sum_{i \in P} y_{ij} + M(1 - \sum_{k \geq j} z_{ik}) \qquad i, j \in P. \qquad \text{(6.9d')}$$

Let $\hat{x}$ be a feasible solution in $\mathcal{T}$ and $\hat{z}$ a fractional vector. Since $C^i x = \sum_{i \in P} y_{ij}$ and $(1 - \sum_{k \geq j} z_{ik})$ is greater than or equal to zero, it is clear that constraint (6.9d') is verified.

□

The above result proves that the linear relaxation of Ω^{GS} is stronger than that of Ω^{θ}, although the two formulations share the same optimal integer values. Nevertheless, as we shall show in the computational experiments, formulation F^{θ} provides much better results in terms of running times and number of optimal solutions found. The reason may be the smaller number of variables used in the second formulation.

To conclude this section we state the relationships between the different formulations that derived from the combination of some OWA representation and any of the STP polytopes described above. To this end, let us denote by $P_{xz}(\Omega^{(\cdot)})$ the projection onto the space of the x, z variables of the linear relaxation of an OWA polytope built on the corresponding $\mathcal{T}^{(\cdot)}$ polytope for the STP. The following property states the relationships among them.

Property 6.3-

$$P_{xz}(\Omega^{sub}) = P_{xz}(\Omega^{km}) = P_{xz}(\Omega^{km2}) \subseteq \begin{cases} P_{xz}(\Omega^{flow}) \\ \neq \\ P_{xz}(\Omega^{mtz}) \end{cases} \tag{6.11}$$

Enhancements and valid inequalities for the OWAP

Formulation F^θ admits other enhancements like removing some redundant variables, adding valid inequalities, etc. First, we observe that since system (6.8b)–(6.8c) contains exactly $2p-1$ linearly independent equations, the above permutation can also be represented without variables z_{i1}, for all $i \in P$, which can be replaced by $1 - \sum_{j \in P: j>1} z_{ij}$. In this way, system (6.8b)–(6.8c) can also be rewritten (see Section 5.3.1) as

$$\sum_{i \in P} z_{ij} = 1 \qquad j \in P : j > 1, \tag{6.12}$$

$$\sum_{j \in P} z_{ij} \leq 1 \qquad i \in P. \tag{6.13}$$

Second, constraints (6.8c) and (6.8e) can be removed from F^θ without changing the set of optimal solutions. We denote by F^θ_{R1} formulation $F^\theta \backslash \{(6.8e)\}$ and by F^θ_{R2} formulation $F^\theta \backslash \{(6.8c), (6.8e)\}$. Note that formulations F^θ_{R1} and F^θ_{R2} admit some solutions that are unfeasible in F^θ (e.g. a solution where $\theta_j \leq \theta_{j+1}$ for some j). However, these two new formulations have fewer constraints and could be more efficient in a branch-and-bound algorithm.

Finally, we present some valid inequalities that can be added to the above OWAP in order to improve the bound of the linear relaxation and/or to reduce the search space in the branch-and-bound tree (see Section 5.4).

- *Constraints related to bounds of cost function values.* Let l_i (u_i) denote the minimum (maximum) objective value relative to cost function $i \in P$, respectively. It is clear that l_i (u_i) are valid lower (upper) bounds on the value of objective i, independently of the position of cost function i in the ordering. Therefore we obtain the following two sets of constraints which are valid for the OWAP:

$$l_i \leq C^i x \leq u_i \qquad i \in P \tag{6.14}$$

- *Constraints related to bounds of values in specific positions.* Let l_j^π (u_j^π) denote the j-th lowest (largest) value of l_i (u_i). Then, l_j^π (u_j^π) is a valid lower (upper) bound of the objective function sorted in position j, that is

$$l_j^\pi \leq \theta_j \leq u_j^\pi \qquad j \in P \tag{6.15}$$

- There are also different bounds on the value of the cost function i and the value of the cost function sorted in position j:

$$\sum_{j \in P} \max\{l_i, l_j^\pi\} z_{ij} \leq C^i x \leq \sum_{j \in P} \min\{u_i, u_j^\pi\} z_{ij} \qquad i \in P \tag{6.16}$$

6.4 Computational experience

Next, we report on the results of some computational experiments we have run, in order to compare empirically the proposed formulations and reinforcements. We have studied the OWASTP combining the different formulations proposed for the STP and the OWAP. First of all we have chosen the best formulation, according to Fernández et al. (2014), among those proposed for the OWAP, namely F_{R2}^z. We recall that the goal of this chapter is the analysis of some STP formulations within the OWAP framework.

For keeping the length of the chapter within some reasonable limits, in our computational experience we study a particular case of the OWA operator, namely the Hurwicz criterion (Hurwicz, 1951), defined as $\alpha \max_{i \in P} y_i + (1 - \alpha) \min_{i \in P} y_i$. This criterion is non-monotonic and non-convex and, in our experience, its behavior in terms of computational effort to get optimal solutions is similar to that of other non-convex OWA criteria. In addition, this objective has been already considered when analyzing the behavior of OWA operators in multiobjective optimization (see, e.g., Galand and Spanjaard, 2012) and it is of special interest for being non-convex since the sorting weights, α, are not in non-increasing order (Grzybowski et al., 2011, Puerto and Tamir, 2005). The number of objectives ranges in $|P| \in \{5, 8, 10\}$ and the considered values of α are $\{0.4, 0.6, 0.8\}$. Graphs are complete according to $|V| \in \{40, 50, 60\}$ and the components of the cost vectors are randomly drawn from a uniform distribution on $[1, 100]$. In addition, for each selection of the parameters $(|V|, p, \alpha)$, 10 instances were randomly generated so, in total, we have a set of 180 benchmark instances. All instances were solved with the MIP Xpress 7.5 optimizer, under a Windows 7 environment in an Intel(R) Core(TM)i7 CPU 2.93 GHz processor and 8 GB RAM. Default values were initially used for all parameters of Xpress solver and a CPU time limit of 3600 seconds was set. We have also tested different combinations of parameters for the solver cut strategy and intensity of heuristics but, unless it is specified, the best results were obtained with the parameters of the solver set to the default values.

In all tables each row summarizes the results corresponding to a group of 10 instances with the same parameters $(|P|, |V|, \alpha)$. Columns are grouped in blocks. The first block contains three columns with the values of the instances parameters. Then, we give a block of four columns for each tested formulation. The columns of each such block are the following. Columns $t(\#)$ report the average computing time in seconds over the 10 instances of the row. If at least one instance reaches the CPU time limit, we indicate in the brackets (#) the number of instances in the group that could be solved to optimality within the CPU time limit. In such a case, t is computed using the CPU time limit for all unsolved instances. Columns $gapLR$ give the percentage relative gap, computed as $100\frac{z^*-z_{LR}}{z_{LR}}$, where z^* denotes the value of the best solution found and z_{LR} the optimal value of the linear relaxation at the root node. Columns gap^* show the maximum percentage optimality gap, over all the instances of the group, relative to the lower bound at termination. Finally, columns *nodes* indicate the average number of nodes explored in the branch-and- bound tree.

The caption just below each block gives the formulation the block refers to. Throughout the section F^{GS} denotes the formulation of Galand and Spanjaard (2012) for the OWASTP. Otherwise, we denote by $F^{(\cdot)}$ the combination of the OWA F^{θ}_{R2} formulation together with a STP $\mathcal{T}^{(\cdot)}$ domain. We report results of formulations F^{GS}, F^{km}, F^{cut}, F^{mtz}, F^{flow}, and F^{km2}. All tables report analogous items for the different formulations described along the chapter. In order to facilitate the comparison among all tables, we have marked in bold red the best result among all in the same group. In case of ties the best results have been marked in bold blue.

6.4.1 Comparison of formulations

In Tables 6.1 and 6.1 we report results for formulations F^{GS}, F^{km}, F^{cut}, F^{mtz}, F^{flow}, and F^{km2}.

The results of block F^{GS} indicate that the OWASTP formulation of Galand and Spanjaard (2012) produces the smallest gaps at the root node ($gapLR$), although only few instances could be solved to optimality, and the gaps remaining at termination (gap^*) are outperformed by most of the other formulations.

The results of block F^{km} show that the number of instances solved to optimality is higher than that of F^{GS}, although the gaps in the unsolved instances are higher. We recall that F^{km} uses $O(n^3)$ variables, which can be too high in large graphs. This also explains the low average number of explored nodes in the B&B tree.

The separation of the cutset inequalities in formulation F^{cut} was implemented using a max-flow based algorithm. Heuristics in Xpress solver were configured with intensity 2 (out of 3) and an initial solution was given to the problem. The initial solution was the minimum cost spanning tree obtained using as edge costs the average costs among all objectives. The results of block F^{cut} indicate that, in general, this formulation outperforms both F^{GS} and F^{km}. As can be seen in block F^{mtz}, the above also holds true for that formulation. In addition, the flow formulation F^{flow} produces a similar performance to the previous one. Observe that the block F^{km2} shows the best performance in terms of both number of instances solved to optimality and running times. The reader may note that the number of explored nodes is significantly smaller than those in previous formulations and the maximum gap at termination (gap^*) is always below 1%.

			F^{GS}				F^{km}				F^{cut}			
$\|P\|$	$\|V\|$	α	$t(\#)$	t^*/gap^*	$gapLR$	$nodes$	$t(\#)$	t^*/gap^*	$gapLR$	$nodes$	$t(\#)$	t^*/gap^*	$gapLR$	$nodes$
5	40	0.4	3303 (1)	14.68%	**41.47**	4530253	3101.6 (2)	48.85%	47.45	13309	350.3	1629.4	48.49	**2809**
5	40	0.6	2432.1 (4)	13.52%	**26.93**	3656732	2365.9 (4)	36.1%	35.95	12585	140.3	490.8	37.37	**1233**
5	40	0.8	2300.5 (4)	3.61%	**8.23**	2028372	1481.3 (7)	19.68%	20.29	5383	685	2806.6	22.24	5846
5	50	0.4	3206.6 (2)	19.38%	**40.38**	3294056	3113.2 (2)	49.63%	47.95	4628	428.7	945	48.83	**1552**
5	50	0.6	3275.6 (1)	13.29%	**27.15**	3557911	3336.6 (1)	37.75%	36.53	4955	820 (**9**)	0.12%	37.53	**2817**
5	50	0.8	3309.7 (2)	1.62%	**7.35**	2043011	2541.9 (5)	20%	20.25	**4469**	1661.1 (8)	0.36%	22.2	5810
5	60	0.4	- (0)	18.25%	**41.59**	1753859	3085.5 (4)	50.31%	48.3	2721	1728.3 (7)	**0.5%**	49.18	**2524**
5	60	0.6	3311.6 (1)	12.23%	**27.43**	1713348	3340.6 (2)	37.27%	36.2	**2125**	1881.4 (6)	**0.29%**	37.78	2714
5	60	0.8	- (0)	4.01%	**8.63**	1662827	- (0)	20.24%	20.27	**4207**	3382.7 (2)	0.88%	22.25	5277
8	40	0.4	- (0)	13.75%	**35.42**	3794485	- (0)	43.13%	41.74	14468	1589.7	3187.7	39.44	**13602**
8	40	0.6	- (0)	12.93%	**25.99**	3724322	- (0)	34.12%	32.16	18424	1727.2 (9)	0.3%	30.25	**15099**
8	40	0.8	- (0)	10.68%	**14.71**	3094546	- (0)	20.12%	19.84	**20398**	2749.9 (5)	0.91%	19.83	21788
8	50	0.4	- (0)	12.86%	**36.45**	1646653	- (0)	46.24%	42.08	**4591**	3021.4 (3)	1.19%	40.43	9660
8	50	0.6	- (0)	25.27%	**27.22**	1772306	- (0)	33.87%	32.14	**5413**	- (0)	1.05%	31.33	11358
8	50	0.8	- (0)	12.64%	**14.85**	1952442	- (0)	20.42%	20.13	**5976**	- (0)	0.85%	20.12	12947
8	60	0.4	- (0)	14.84%	**36.47**	1417232	- (0)	48.98%	41.29	**1889**	- (0)	1.02%	39.56	5801
8	60	0.6	- (0)	25.98%	**27.21**	1472262	- (0)	34.81%	32.09	**2279**	- (0)	0.9%	30.77	6144
8	60	0.8	- (0)	13.74%	**15.74**	1347127	- (0)	20.68%	19.84	**2555**	- (0)	0.74%	20.14	6767

Table 6.1: OWASTP results for the different formulations.

			F^{mtz}				F^{flow}				F^{km2}			
$\|P\|$	$\|V\|$	α	$t(\#)$	t^*/gap^*	$gapLR$	$nodes$	$t(\#)$	t^*/gap^*	$gapLR$	$nodes$	$t(\#)$	t^*/gap^*	$gapLR$	$nodes$
5	40	0.4	161	1084.4	48.09	330936	108	707.3	48.47	132275	**37.8**	**289.7**	47.25	6005
5	40	0.6	65.8	281.2	36.87	134268	**19.1**	**46.1**	37.34	27210	21.1	115.3	35.86	3525
5	40	0.8	40.2	165.3	21.63	55925	82.8	371	22.21	79107	**11.3**	**30.7**	20.37	**2530**
5	50	0.4	1250.3 (8)	1.58%	48.53	1719405	239	**815.5**	48.77	186992	**202.6**	1385.5	47.63	22038
5	50	0.6	1292.9 (7)	2.41%	37.28	1780642	772.1 (**9**)	**0.06%**	37.47	598456	**510.7** (**9**)	0.85%	36.09	42928
5	50	0.8	1010.2 (8)	0.5%	21.65	880870	**511.8**	**2560**	22.11	263337	586.6 (9)	0.49%	20.37	78392
5	60	0.4	3481.3 (1)	8.72%	50.56	2151974	1469.8 (7)	0.98%	49.19	689822	**1107.2 (8)**	0.62%	47.98	39445
5	60	0.6	3381.2 (1)	7.74%	39.06	2399936	1334.1 (7)	0.48%	37.75	562257	**736.5 (9)**	0.3%	36.31	27597
5	60	0.8	3540.9 (1)	4.62%	22.78	2495350	2822.4 (4)	0.89%	22.22	998333	**1790.2 (6)**	**0.43%**	20.37	136451
8	40	0.4	1444.6 (8)	3.26%	39.37	2447077	679.2 (9)	1.16%	39.43	493769	**144.5**	**598.6**	38.55	63802
8	40	0.6	1540.2 (7)	0.6%	30.04	2555899	717.7 (9)	0.42%	30.22	513069	**115.7**	**654.2**	29.22	68677
8	40	0.8	1837.4 (6)	0.84%	19.54	2436963	1366.7 (8)	0.86%	19.78	848285	**107**	**496.6**	18.61	61329
8	50	0.4	3290.4 (1)	4.22%	41.2	2825835	3021.2 (2)	1.93%	40.52	1312489	**1093.3 (8)**	**0.99%**	39.67	232801
8	50	0.6	3491.4 (1)	4.25%	31.97	3094747	2897.2 (4)	2.37%	31.43	1505621	**1505.5 (8)**	**0.38%**	30.41	569556
8	50	0.8	2781.1 (4)	1.71%	20.04	2254217	2352.8 (5)	**0.83%**	20.08	851597	**1559.9 (7)**	0.96%	19.12	533160
8	60	0.4	- (0)	12.09%	43.53	1374362	3310.1 (1)	3.54%	39.8	1350196	**1315 (9)**	**0.51%**	38.52	294879
8	60	0.6	- (0)	7.96%	33.49	1891401	- (0)	2.28%	30.96	1316337	**1164.5 (8)**	**0.52%**	29.63	246070
8	60	0.8	- (0)	4.75%	21.54	2104458	- (0)	**0.69%**	20.09	1086016	**1688.5 (8)**	0.7%	18.86	446946

Table 6.2: OWASTP results for the different formulations.

The previous results show that formulation F^{km2} outperforms all the others. Thus, we have tested several reinforcements of this formulation to further improve its performance. Next we report on the three most promising strengthening found, which consist on adding valid inequalities (6.14), (6.15) and (6.16) to formulation F^{km2}. Tables 6.3 and 6.4 show that the performance of F^{km2} is highly improved whenever any of these valid inequalities are added. In all cases at most 2 instances could not be solved to optimality within the CPU time limit, and in those cases the remaining gaps at termination were quite small (below 0.5%).

Table 6.5 displays a comparison among the results obtained by Galand and Spanjaard, 2012 for the OWASTP and our best formulation. First, column F^{GS} shows the formulation implemented by Galand and Spanjaard, 2012 in IBM ILOG CPLEX 12 without any preprocessing whereas columns F^{GS}_{P1} and F^{GS}_{P2} shows the running times after applying two different preprocessings (shavings) described in that paper. In that case, 30 instances were considered for each tuple $(|P|, |V|, \alpha)$ and symbol "-" indicates

$\|P\|$	$\|V\|$	α	F^{km2}: $t(\#)$	t^*/gap^*	$gapLR$	$nodes$	F^{km2} + (6.14): $t(\#)$	t^*/gap^*	$gapLR$	$nodes$
5	40	0.4	37.8	289.7	47.25	6005	18	95.7	47.22	2869
5	40	0.6	21.1	115.3	35.86	3525	**9.2**	**20.8**	35.83	**873**
5	40	0.8	11.3	30.7	20.37	2530	9.2	17.9	20.32	2160
5	50	0.4	202.6	1385.5	47.63	22038	**73.8**	**592.1**	47.56	**12474**
5	50	0.6	510.7 (9)	0.85%	36.09	42928	**18.9**	**39.3**	35.98	**1124**
5	50	0.8	586.6 (9)	0.49%	20.37	78392	**16.6**	**31.6**	20.24	2575
5	60	0.4	1107.2 (8)	0.62%	47.98	39445	82.8	543.3	47.93	3768
5	60	0.6	736.5 (9)	0.3%	36.31	27597	**168**	**1144.8**	36.23	17012
5	60	0.8	1790.2 (6)	0.43%	20.37	136451	512.3 (9)	0.29%	20.27	54782
8	40	0.4	144.5	598.6	38.55	63802	45.5	201.8	38.59	19952
8	40	0.6	115.7	654.2	29.22	68677	**21**	47.1	29.25	**8453**
8	40	0.8	107	496.6	18.61	61329	42	90.4	18.58	21314
8	50	0.4	1093.3 (8)	0.99%	39.67	232801	108.9	480.5	39.6	39815
8	50	0.6	1505.5 (8)	**0.38**%	30.41	569556	**437.1** (**9**)	0.46%	30.34	157305
8	50	0.8	1559.9 (7)	0.96%	19.12	533160	166.5	**321.7**	18.97	56487
8	60	0.4	1315 (9)	0.51%	38.52	294879	**229**	981.9	38.44	67639
8	60	0.6	1164.5 (8)	0.52%	29.63	246070	437.8	1037.9	29.54	162671
8	60	0.8	1688.5 (8)	0.7%	18.86	446946	677.8	1838.6	18.74	200817

Table 6.3: OWASTP results for F^{km2} including valid inequalities

$\|P\|$	$\|V\|$	α	F^{km2} + (6.15): $t(\#)$	t^*/gap^*	$gapLR$	$nodes$	F^{km2} + (6.16): $t(\#)$	t^*/gap^*	$gapLR$	$nodes$
5	40	0.4	15.6	62.9	**36.51**	**1525**	**12.1**	**40.1**	47.22	1821
5	40	0.6	11.6	27.6	**30.04**	1085	12.2	33	35.83	1480
5	40	0.8	9.2	19	**17.63**	**1130**	**8.7**	**17.2**	20.32	1301
5	50	0.4	204.2	1871.5	**37.65**	30819	273.4	2564.3	47.56	31749
5	50	0.6	26.4	60.8	**30.61**	1746	42.3	267.4	35.98	2753
5	50	0.8	18.4	37.1	**17.73**	**1927**	21.4	63.1	20.24	2330
5	60	0.4	73.9	339.2	**38.86**	3053	**37.4**	**116.1**	47.93	**1301**
5	60	0.6	312.6	2622.2	**31.29**	16891	270.8	2019.6	36.23	**13980**
5	60	0.8	**479.7** (9)	0.29%	**17.96**	**33318**	529.3	**3253**	20.27	60893
8	40	0.4	78.1	389.2	**29.4**	18236	**36.3**	**103.4**	38.59	**12104**
8	40	0.6	41.5	149.9	**24.54**	10984	23.6	**45.2**	29.25	8565
8	40	0.8	64.2	144.2	**16.55**	20397	**41.4**	**78.4**	18.58	**19360**
8	50	0.4	125.7	408.9	**31.31**	**26170**	**89.7**	**271.5**	39.6	27126
8	50	0.6	468.5 (**9**)	0.4%	**26.09**	**115520**	438.2 (**9**)	0.4%	30.34	129181
8	50	0.8	217.9	436	**17.12**	**47529**	**147.4**	370.9	18.97	52554
8	60	0.4	370	1550.8	**30.67**	**60953**	249	**942.9**	38.44	65411
8	60	0.6	617.5	1569.8	**25.59**	102501	**351.3**	**853.8**	29.54	**97160**
8	60	0.8	1083	3356.9	**17.03**	199707	**489**	**1090.4**	18.74	**160118**

Table 6.4: OWASTP results for F^{km2} including valid inequalities

that the average execution times is beyond 15min (1800s). Only minimum, average and maximum running times were provided. Our results in column F^{km2} + (6.16) show a better performance in mostly all cases.

$\|P\|$	$\|V\|$	α	F^{GS}: t_*	t	t^*	F^{GS}_{P1}: t_*	t	t^*	F^{GS}_{P2}: t_*	t	t^*	F^{km2} + (6.16): t_*	t	t^*
5	20	0.4	**0.3**	1.2	**2.3**	0.5	1.4	2.6	**0.3**	**1.1**	**2.3**	1.1	1.6	3.6
5	20	0.6	0.9	1.8	3.4	1.1	2.1	3.4	**0.6**	1.7	2.8	1	**1.4**	**2.4**
5	20	0.8	0.6	1.7	4.3	**0.5**	1.5	3.4	**0.5**	1.5	3.1	0.9	**1.2**	**1.8**
5	30	0.4	2.4	**4.6**	**10.1**	3.6	6.1	11.8	**2.2**	**4.6**	**10.1**	2.8	5.8	16.1
5	30	0.6	1.9	**8.9**	**41.7**	3.1	10.3	44.3	**1.6**	9.1	43.8	2.4	14.4	72.9
5	30	0.8	1.5	28.1	104.7	1.2	18.6	**60.7**	**0.5**	18.4	90.5	1.9	**15.5**	75
5	40	0.4	6.5	18	50.8	11.5	23.9	57	6.6	18.1	45.3	**5.4**	**12.1**	**40.1**
5	40	0.6	7.9	46.2	155.8	13.3	51.9	163.6	7.7	46.1	155.6	**5.1**	**12.2**	**33**
5	40	0.8	6.5	70.5	211.7	7.1	53.4	184.8	**4.1**	51.7	226	4.2	**8.7**	**17.2**
5	50	0.4	21.7	**123.9**	**323.6**	38.8	143.4	352.7	21.5	124.6	335.8	**8.3**	273.4	2564.3
5	50	0.6	26.3	367.3	2374.1	41.7	384.8	2404.1	26	368	2394.3	**7.6**	**42.3**	**267.4**
5	50	0.8	**9.7**	297.8	3664.5	23.3	217.6	1972.1	14.5	225.3	1978.9	10	**21.4**	**63.1**
5	60	0.4	41	460.9	4131.3	86.9	511.7	4174.9	40.5	461.1	4092.9	**18.8**	**37.4**	**116.1**
5	60	0.6	-	-	-	-	-	-	-	-	-	**12.7**	**270.8**	**2019.6**
5	60	0.8	6.4	1725.3	16778.8	25.2	866.9	9426.5	**2.6**	1586.2	29575.1	18.7	**529.3**	**3253**

Table 6.5: Comparison among the results obtained by Galand and Spanjaard, 2012

Finally we run a last series of experiments with larger graphs of sizes up to 100 nodes and with up to 10 objectives. To the best of our knowledge the largest OWASTP instances reported in the literature have up to 60 nodes and up to 5 objectives.

Table 6.6 shows our results of formulations $F^{km2}+(6.15)$ and $F^{km2}+(6.16)$. We can observe that, when $|V| \geq 80$, after 1h of CPU time there are already some unsolved instances. Nevertheless, the performance of $F^{km2}+(6.15)$ and $F^{km2}+(6.16)$ is remarkable, as the biggest gaps at termination are around 1% and always below 5%.

$\|P\|$	$\|V\|$	α	$t(\#)$	t^*/gap^*	$gapLR$	$nodes$	$t(\#)$	t^*/gap^*	$gapLR$	$nodes$
5	40	0.4	15.6	62.9	**36.51**	**1525**	**12.1**	**40.1**	47.22	1821
5	40	0.6	**11.6**	**27.6**	**30.04**	**1085**	12.2	33	35.83	1480
5	40	0.8	9.2	19	**17.63**	**1130**	**8.7**	**17.2**	20.32	1301
5	50	0.4	**204.2**	**1871.5**	**37.65**	**30819**	273.4	2564.3	47.56	31749
5	50	0.6	**26.4**	**60.8**	**30.61**	**1746**	42.3	267.4	35.98	2753
5	50	0.8	**18.4**	**37.1**	**17.73**	**1927**	21.4	63.1	20.24	2330
5	60	0.4	73.9	339.2	**38.86**	3053	**37.4**	**116.1**	47.93	**1301**
5	60	0.6	312.6	2622.2	**31.29**	16891	**270.8**	**2019.6**	36.23	**13980**
5	60	0.8	**479.7** (9)	0.29%	**17.96**	**33318**	529.3	**3253**	20.27	60893
5	80	0.4	1460.2 (**8**)	1.2%	**39.8**	**14117**	**1193.2** (**8**)	**1**%	47.48	19421
5	80	0.6	**1682.5** (**7**)	**1.09**%	**32.02**	22844	1828.2 (**7**)	1.49%	36.15	**22162**
5	80	0.8	**1554.2** (**6**)	**1.09**%	**18.41**	**21643**	1583 (**6**)	1.29%	20.38	28128
5	100	0.4	**2241** (**5**)	1.15%	**40.59**	**12791**	2336 (**5**)	**1.14**%	47.88	16091
5	100	0.6	2269.4 (4)	**1.17**%	**32.33**	13356	**2191.9** (**5**)	4.48%	36.56	**9957**
5	100	0.8	1591.5 (**6**)	1.52%	**18.38**	**19327**	**1548.9** (**6**)	**0.85**%	20.21	29653
8	40	0.4	78.1	389.2	**29.4**	18236	**36.3**	**103.4**	38.59	**12104**
8	40	0.6	41.5	149.9	**24.54**	10984	**23.6**	**45.2**	29.25	**8565**
8	40	0.8	64.2	144.2	**16.55**	20397	**41.4**	**78.4**	18.58	**19360**
8	50	0.4	125.7	408.9	**31.31**	**26170**	**89.7**	**271.5**	39.6	27126
8	50	0.6	468.5 (**9**)	**0.4**%	**26.09**	**115520**	438.2 (**9**)	**0.4**%	30.34	129181
8	50	0.8	217.9	436	**17.12**	**47529**	**147.4**	**370.9**	18.97	52554
8	60	0.4	370	1550.8	**30.67**	**60953**	**249**	**942.9**	38.44	65411
8	60	0.6	617.5	1569.8	**25.59**	102501	**351.3**	**853.8**	29.54	**97160**
8	60	0.8	1083	3356.9	**17.03**	199707	**489**	**1090.4**	18.74	**160118**
8	80	0.4	2314 (**5**)	1.46%	**35.14**	**105903**	**2176.4** (**5**)	**1.02**%	38.59	152486
8	80	0.6	2785.9 (**5**)	**0.2**%	**26.33**	**215590**	**2355.2** (**5**)	0.5%	29.65	316561
8	80	0.8	3085.1 (3)	1.16%	**17.44**	**234384**	**2010.7** (**8**)	**0.98**%	18.85	299480
8	100	0.4	3238 (3)	3.82%	**33.18**	**90102**	**2903.9** (**4**)	**1.11**%	38.82	119588
8	100	0.6	**2952.3** (**3**)	0.81%	**26.79**	**81007**	2998.1 (**3**)	**0.58**%	29.73	153248
8	100	0.8	3595.7 (**1**)	1.23%	**17.71**	**159736**	**3396.8** (**1**)	**0.93**%	19	232543
10	40	0.4	1360.3 (7)	0.05%	**27.04**	139960	**263.8**	**837.5**	35.82	**128102**
10	40	0.6	1345.1 (**8**)	1.25%	**22.82**	**336561**	**986.5** (**8**)	**1**%	27.25	435060
10	40	0.8	1980.9 (7)	**0.66**%	**15.62**	**557424**	**1054.3** (**9**)	1.34%	17.55	589547
10	50	0.4	2209.9 (5)	1.31%	**28.44**	**311753**	**1679.6** (6)	**1.12**%	36.01	494558
10	60	0.4	- (0)	**0.51**%	28.5	**373784**	2604.4 (4)	**0.85**%	**27.53**	863090
10	60	0.6	- (0)	**0.62**%	23.54	**469722**	**2798.2** (4)	0.9%	**18.08**	1097300
10	60	0.8	- (0)	0.63%	**16.22**	**478948**	**2135.2** (**7**)	**0.44**%	35.72	600445
10	80	0.4	- (0)	2.76%	28.96	**176553**	**2625.9** (**7**)	**0.58**%	27.17	934221
10	80	0.6	- (0)	0.88%	23.99	**203533**	**3514.7** (1)	**0.54**%	**17.74**	996344
10	80	0.8	- (**0**)	1.02%	**16.45**	**213147**	- (**0**)	0.91%	34.99	467169
10	100	0.4	- (**0**)	1.02%	29.51	**108833**	- (**0**)	**0.47**%	27.03	458488
10	100	0.6	- (**0**)	**0.65**%	24.12	**103042**	- (**0**)	**0.9**%	**17.72**	492075
10	100	0.8	- (**0**)	1.27%	**16.53**	**101222**	- (**0**)	0.84%	35.01	222124
			kp21_1_3_cota_j				kp21_1_3_cota_y_z			

Table 6.6: OWASTP results for large instances

6.5 Conclusions

In this chapter we have presented reinforced mathematical programming formulations for the OWASTP as well as a new formulation which reduces the number of decision variables. This new formulation reinforced with appropriate constraints has shown to be very promising for efficiently solving many medium size OWASTP instances. However, from the obtained results it is also clear that for solving larger OWASTP instances with more objective functions further improvements are needed. Our current research focuses on the design of more sophisticated elimination tests as well as from alternative formulations leading to tighter LP bounds.

Bibliography

Andersen, K., Jörnsten, K. and Lind, M. (1996) "On bicriterion minimal spanning trees: an approximation". *Computers & Operations Research*, vol. 23 (12): 1171–1182.

Averbakh, I. (2001) "On the complexity of a class of combinatorial optimization problems with uncertainty". *Mathematical Programming*, vol. 90: 263–272.

Averbakh, I., Berman, O. and Punnen, A. (1995) "Constrained matroidal bottleneck problems". *Discrete Applied Mathematics*, vol. 63(3): 201–214.

Balas, E. and Padberg, M. (1972) "On the set-covering problem". *Operations Research*, vol. 20: 1152–1161.

Boland, N., Domínguez-Marín, P., Nickel, S. and Puerto, J. (2006) "Exact procedures for solving the discrete ordered median problem". *Computers & Operations Research*, vol. 33 (11): 3270–3300.

Breuer, M. A. (1970) "Simplification of the covering problem with application to Boolean expressions". *J. Assoc. Comput. Mach*, vol. 17: 166–181.

Calvete, H. and Mateo, P. (1998) "Lexicographic optimization in generalized network flow problems". *Journal of Operations Research Society*, vol. 49(2): 519–529.

Christofides, N. and Korman, S. (1974) "A computational survey of methods for the set covering problem". *Management Science*, vol. 21(5): 591–599.

Conde, E. (2004) "An improved algorithm for selecting p items with uncertain returns according to the minmax-regret criterion". *Mathematical Programming*, vol. 100: 345–353.

Croce, F. D., Paschos, V. T. and Tsoukias, A. (1999) "An improved general procedure for lexicographic bottleneck problems". *Operations Research Letters*, vol. 24: 187–194.

Edmonds, J. (1970) *Submodular functions, matroids, and certain polyhedra*, pp. 69–87. Gordon and Breach, New York.

Edmonds, J. (1971) "Matroids and the greedy algorithm". *Mathematical Programming*, vol. 1: 127–136.

Ehrgott, M. (2005) *Multicriteria Optimization*. Springer Berlin Heidelberg.

Espejo, I., Marin, A., Puerto, J. and Rodríguez-Chía, A. (2009) "A comparison of formulations and solution methods for the minimum-envy location problem". *Computers & Operations Research*, vol. 36: 1966–1981.

Fernández, E., Pozo, M. A. and Puerto, J. (2014) "Ordered weighted average combinatorial optimization: Formulations and their properties". *Discrete Applied Mathematics*, vol. 169(0): 97–118.

Fernández, E., Pozo, M. A. P. and Puerto, J. (2013) "A modeling framework for ordered weighted average combinatorial optimization". *arXiv:1306.1426*.

Galand, L. and Perny, P. (2007) "Search for choquet-optimal paths under uncertainty". In *Proceedings of the 23rd conference on uncertainty in artificial intelligence, AAAI Press*, pp. 125–132.

Galand, L. and Spanjaard, O. (2012) "Exact algorithms for OWA-optimization in multiobjective spanning tree problems". *Computers & Operations Research*, vol. 39: 1540–1554.

Galil, Z. and Schieber, B. (1998) "On finding most uniform spanning trees". *Discrete Applied Mathematics*, vol. 20: 173–175.

Garfinkel, R., Fernández, E. and Lowe, T. (2006) "The k-centrum shortest path problem". *TOP*, vol. 14: 279–292.

Gavish, B. (1983) "Formulations and algorithms for the capacitated minimal directed tree problem". *Journal of the ACM*, vol. 30: 118–132.

Gouveia, L. (1995) "Using the miller-tucker-zemlin constraints to formulate a minimal spanning tree problem with hop constraints". *Computers & Operations Research*, vol. 22(9): 959–970.

Grzybowski, J., Nickel, S., Pallaschke, D. and Urbanski, R. (2011) "Ordered median functions and symmetries". *Optimization*, vol. 60(7): 801–811.

Gupta, M., Gupta, Y. and Bector, C. (1990) “Minimizing the flow-time variance in single-machine systems”. *Journal of Operations Research Society*, vol. 41(8): 769–779.

Gupta, S. K. and Punnen, A. P. (1988) “Minimum deviation problems”. *Operations Research Letters*, vol. 7(4): 201–204.

Hamacher, H. and Ruhe, G. (1994) “On spanning tree problems with multiple objectives”. *Annals of Operations Research*, vol. 52: 209–230.

Hansen, P. (1980) “Bicriterion path problems”. In *Multiple criteria decision making theory and application*, vol. 177 of Lecture Notes in Econom. and Math. Systems, pp. 109–127.

Hansen, P. and Labbé, M. (1988) “Algorithms for voting and competitive location on a network”. *Transportation Science*, vol. 22(4): 278–288.

Hansen, P., Labbé, M. and Thisse, J. (1991) “From the median to the generalized center.” *RAIRO, Rech. Opér.*, vol. 25: 73–86.

Hu, T. (1974) “Optimum communication spanning trees”. *SIAM Journal on Computing*, vol. 3: 188–195.

Hurwicz, L. (1951) “Optimality criteria for decision making under ignorance”. *Cowles Commission Discussion Paper, Chicago*, p. 370.

Kalcsics, J., Nickel, S., Puerto, J. and Tamir., A. (2002) “Algorithmic results for ordered median problems defined on networks and the plane”. *Operations Research Letters*, vol. 30: 149–158.

Kasperski, A. and Zielinski, P. (2013) “Combinatorial optimization problems with uncertain costs and the OWA criterion”. *arXiv:1305.5339*.

Kelly, L. M. (1944) “Covering problems”. *National Mathematics Magazine*, vol. 19: 123–130.

Kruskal, J. (1956) “On the shortest spanning subtree of a graph and the traveling salesman problem”. *Proceedings of the American Mathematical Society*, vol. 7: 48–50.

Landete, M. and Marín, A. (2014) “Looking for edge-equitable spanning trees”. *Computers & Operations Research*, vol. 41(0): 44–52.

Laporte, G. (1992) “The traveling salesman problem: An overview of exact and approximate algorithms”. *European Journal of Operational Research*, vol. 59(2): 231–247.

Lawler, E. L. (1966) “Covering problems: Duality relations and a new method of solution”. *SIAM Journal on Applied Mathematics*, vol. 14: 1115–1132.

Lawler, E. L. (1972) “A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem”. *Management Science*, vol. 18(7): 401–405.

López de los Mozos, M., Mesa, J. and Puerto, J. (2008) “A generalized model of equality measures in network”. *Computers & Operations Research*, vol. 35: 651–660.

Magnanti, T. L. and Wolsey, L. A. (1995) “Optimal trees”. In C. M. M.O. Ball, T.L. Magnanti and G. Nemhauser, eds., *Network Models, Chapter 9*, vol. 7 of *Handbooks in Operations Research and Management Science*, pp. 503 – 615. Elsevier.

Marín, A., Nickel, S., Puerto, J. and Velten, S. (2009) “A flexible model and efficient solution strategies for discrete location problems”. *Discrete Applied Mathematics*, vol. 157(5): 1128–1145.

Martello, S., Pulleyblank, W., Toth, P. and de Werra, D. (1984) “Balanced optimization problems”. *Operations Research Letters*, vol. 3(5): 275–278.

Martin, R. (1991) “Using separation algorithms to generate mixed integer model reformulations”. *Operations Research Letters*, vol. 10(3): 119–128.

Mesa, J., Puerto, J. and Tamir, A. (2003) “Improved algorithm for several network location problems with equality measures”. *Discrete Applied Mathematics*, vol. 130: 437–448.

Miller, C. E., Tucker, A. W. and Zemlin, R. A. (1960) “Integer programming formulation of traveling salesman problems”. *Journal of the ACM*, vol. 7(4): 326–329.

Minoux, M. (1989) "Network synthesis and optimum network design problems: models, solution methods and applications". *Networks*, vol. 19: 313–360.

Ogryczak, W. and Olender, P. (2012) "On MILP models for the OWA optimization". *Journal of Telecommunications and Information Technology*, vol. 2/2012: 5–12.

Ogryczak, W. and Sliwinski, T. (2003) "On solving linear programs with the ordered weighted averaging objective". *European Journal of Operational Research*, vol. 148(1): 80–91.

Ogryczak, W. and Tamir, A. (2003) "Minimizing the sum of the k largest functions in linear time". *Information Processing Letters*, vol. 85 (3): 117–122.

Pascoal, M., Captivo, M. E. and Clímaco, J. (2003) "A note on a new variant of Murty's ranking assignments algorithm". *4OR*, vol. 1(3): 243–255.

Prim, R. (1957) "Shortest connection networks and some generalizations". *Bell System Technical Journal*, vol. 36: 1389–1401.

Puerto, J. (2008) "A new formulation of the capacitated discrete median problem with $\{0,1\}$-assignment". In J. Kalcsics and S. Nickel, eds., *Operations Research Proceedings 2007*, chap. , pp. 165–170. Springer, Heidelberg.

Puerto, J. and Rodríguez-Chía, A. (2003) "Robust positioning of service units". *Stochastic Models*, vol. 19(1): 125–147.

Puerto, J. and Tamir, A. (2005) "Locating tree-shaped facilities using the ordered median objective". *Mathematical Programming*, vol. 102(2): 313–338.

Punnen, A. (1992) "Traveling salesman problem under categorization". *Operations Research Letters*, vol. 12(2): 89–95.

Punnen, A., Nair, K. and Aneja, Y. (1995) "Generalized bottleneck problems". *Optimization*, vol. 35(2): 159–169.

Punnen, A. P. and Aneja, Y. P. (1997) "Minimum dispersion problems". *Discrete Applied Mathematics*, vol. 75(1): 93–102.

Schrijver, A. (1983) *Min-max results in combinatorial optimization*, pp. 439–500. Springer, Berlin.

Slater, P. (1978a) "Centers to centroids in graphs". *Journal of Graph Theory*, vol. 2(3): 209–222.

Slater, P. (1978b) "Structure of the k-centra in a tree". In *Proceedings of the Ninth Southeastern Conference on Combinatorics, Graph Theory, and Computing*, vol. XXI, pp. 663–670.

Sourd, F. and Spanjaard, O. (2008) "A multi-objective branch-and-bound framework. application to the bi-objective spanning tree problem". *INFORMS Journal of Computing*, vol. 20 (3): 472–484.

Steiner, S. and Radzik, T. (2008) "Computing all efficient solutions of the biobjective minimum spanning tree problem". *Computers & Operations Research*, vol. 35 (1): 198–211.

Tamir, A. (2000) "The k-centrum multi-facility location problem". *Dis*, vol. 109: 292–307.

Tamir, A., Puerto, J. and Perez-Brito, D. (2002) "The centdian subtree on tree networks". *Discrete Applied Mathematics*, vol. 118: 263–278.

Yen, J. Y. (1971) "Finding the K shortest loopless paths in a network". *Management Science*, vol. 17: 712–716.

Yu, G. (1998) "Min-max optimization of several classical discrete optimization problems." *Journal of Optimization Theory and Applications*, vol. 98(1): 221–42.

Conclusions

This PhD dissertation explores different topics related to mathematical models for the design and planning of transportation on demand in urban logistics networks. The contributions are divided into six main chapters and a basic background, for the contents that are presented, is provided.

In Chapter 1, a new approach for jointly planning timetables and vehicle schedules along a single transit line has been developed by emphasizing the point of view of potential customers. The setting analyzed in this chapter assumes a model of fully disaggregated demand for a scenario that includes capacity constraints and demand behavior according to different criteria. A p-median based formulation has been proposed including specific constraints for the scheduling problem for a given fleet size of vehicles. In addition, demand behavior is associated with the inclusion of closest assignment type constraints. A clustering algorithm has been developed in order to provide an alternative methodology for solving instances of the problem when computational time must be limited. The performed computational experience shows the difficulty of including closest assignment constraints in a transportation problem and the advantages of deriving a clustering algorithm that allows an appropriate preprocessing of the information.

In Chapter 2, we have presented a new approach for solving the integration of the Transit Network Timetabling and Scheduling Problem together with the passengers' routing problem. Traditionally, these problems have been studied sequentially but this approach leads to sub-optimal solutions for the entire problem. We present a flexible framework that let us allocate transportation requests to their optimal strategies under capacity constraints. This approach not only pursues transfer coordination but also customers' preferences in terms of preferred departure/arrival times for a fully disaggregated demand. Even more, each transportation request is faced individually, stating hard time windows constraints for departure/arrival times as well as inconvenience costs related to trip duration and time deviations from desired departure/arrival times. A testbed of randomly generated instances has been generated for different network configurations existant in the literature and computational results have been obtained and analyzed.

In Chapter 3, we have presented a modeling approach for solving the rescheduling problem in a transit line that has suffered a fleet size reduction. We have described a demand pattern to reflect the passengers' behaviour when some vehicle services are delayed or cancelled. This inconvenience function has been used to derive a rescheduling framework coming from a timetabling formulation. We have shown that the problem can be solved rapidly by using a constrained max-cost-flow problem whose coefficient matrix we prove is totally unimodular. We have tested the different formulations over a testbed of random instances and the results show that (1) on-line rescheduling can be efficiently done by using the proposed models, (2) previous approaches in the literature are outperformed and (3) our approach can be applied to real scenarios as it is the case of the commuter train system of Madrid.

In Chapter 4, we have provided a methodology to obtain a complete description of the set of Pareto-optimal solutions for the multi-criteria p-facility median location problem on networks. It is noteworthy that this chapter is the first attempt to characterize the solution set of this problem. Note that the single criteria p-facility median problem is already NP-hard and handling closest assignments makes more difficult to deal with the multifacility version. The main tools used to obtain the set of Pareto-optimal solutions is the characterization of the linearity domains of the distance functions and the lower envelope. Hence, this analysis can be easily extended to more general objective functions as long as we can again determine these domains and their image and preimage.

In Chapter 5, we have presented and revisited different mathematical programming formulations for the OWAP using different sets of decision variables. These formulations reinforced with appropriate constraints have shown to be rather promising for efficiently solving many medium size OWAP instances.

Finally, in Chapter 6 we have studied the Ordered Weighted Average Spanning Tree Problem. The ordered weighted average is an averaging operator to aggregate the vector of objective values of feasible solutions. A new complexity result is proven according to which the OWASTP is NP-complete even for cactus graphs and two objectives. Alternative mixed integer linear formulations have been proposed and compared, both theoretically and empirically. Extensive computational experiments on a large set of randomly generated benchmark instances have been run and the obtained numerical results analyzed and compared. These results show that the choice of an appropriate formulation allows to solve larger instances with more objectives than those previously solved in the literature.

List of Figures

List of Tables

Bibliography

Altay, N. and Green, W. G. (2006) "OR/MS research in disaster operations management". *European Journal of Operational Research*, vol. 175(1): 475–493.

Andersen, K., Jörnsten, K. and Lind, M. (1996) "On bicriterion minimal spanning trees: an approximation". *Computers & Operations Research*, vol. 23 (12): 1171–1182.

Arriola, R., Ortega, F. A. and Pozo, M. A. (2009) "Aplicación de los problemas pickup and delivery en la gestión de emergencias de un sistema de ferrocarril". In *Actas del II MORE (II Jornadas sobre Modelos de Optimización Aplicados a la Planificación Robusta y a la Gestión de los Servicios de Transporte en Caso de Emergencia)*, vol. 1, pp. 55–64. Universidad de Sevilla.

Averbakh, I. (2001) "On the complexity of a class of combinatorial optimization problems with uncertainty". *Mathematical Programming*, vol. 90: 263–272.

Averbakh, I., Berman, O. and Punnen, A. (1995) "Constrained matroidal bottleneck problems". *Discrete Applied Mathematics*, vol. 63(3): 201–214.

Balas, E. and Padberg, M. (1972) "On the set-covering problem". *Operations Research*, vol. 20: 1152–1161.

Bitran, G. and Lawrence, K. (1980) "Location service offices: a multicriteria approach". *Omega*, vol. 8(2): 201–206.

Boland, N., Domínguez-Marín, P., Nickel, S. and Puerto, J. (2006) "Exact procedures for solving the discrete ordered median problem". *Computers & Operations Research*, vol. 33 (11): 3270–3300.

Brannlund, U., Lindberg, P. O., Nou, A. and Nilsson, J.-E. (1998) "Railway timetabling using Lagrangian relaxation". *Transportation Science*, vol. 32(4): 358–369.

Breuer, M. A. (1970) "Simplification of the covering problem with application to Boolean expressions". *J. Assoc. Comput. Mach*, vol. 17: 166–181.

Bunte, S. and Kliewer, N. (2009) "An overview on vehicle scheduling models". *Public Transport*, vol. 1: 299–317.

Burdett, R. and Kozan, E. (2009) "Techniques for inserting additional trains into existing timetables". *Transportation Research Part B: Methodological*, vol. 43(8–9): 821–836.

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L. and Wagenaar, J. (2014) "An overview of recovery models and algorithms for real-time railway rescheduling". *Transportation Research Part B: Methodological*, vol. 63(0): 15–37.

Cacchiani, V. and Toth, P. (2012) "Nominal and robust train timetabling problems". *European Journal of Operational Research*, vol. 219(3): 727–737. Feature Clusters.

Cadarso, L. and Marín, A. (2012) "Integration of timetable planning and rolling stock in rapid transit networks". *Annals of Operations Research*, vol. 199(1): 113–135.

Cai, X. and Goh, C. J. (1994) "A fast heuristic for the train scheduling problem". *Computers & Operations Research*, vol. 21(5): 499–510.

Calvete, H. and Mateo, P. (1998) "Lexicographic optimization in generalized network flow problems". *Journal of Operations Research Society*, vol. 49(2): 519–529.

Caprara, A., Fischetti, M. and Toth, P. (2002) "Modeling and solving the train timetabling problem". *Operations Research*, vol. 50(5): 851–861.

Caprara, A., Kroon, L., Monaci, M., Peeters, M. and Toth, P. (2007) *Passenger railway optimization.*, vol. 14, pp. 129–187. Elsevier, Amsterdam.

Caprara, A., Kroon, L. and Toth, P. (2011) *Optimization problems in passenger railway systems*, vol. 6, pp. 3896–3905. Wiley.

Caprara, A., Monaci, M., Toth, P. and Guida, P. L. (2006) "A Lagrangian heuristic algorithm for a real-world train timetabling problem". *Discrete Applied Mathematics*, vol. 154(5): 738–753.

Carrano, E., Takahashi, R., Finseca, C. and Neto, O. (2007) "Bi-objective combined facility location network design". *EMO 2007, LNCS*, vol. 4403: 486–500.

Carrizosa, E., Conde, E., Fernández, F. and Puerto, J. (1993) "Efficiency in euclidean constrained location problems." *Operations Research Letters*, vol. 14: 291–295.

Castelli, L., Pesenti, R. and Ukovich, W. (2004) "Scheduling multimodal transportation systems". *European Journal of Operational Research*, vol. 155(3): 603–615.

Ceder, A. (2001) *Efficient timetabling and vehicle scheduling for public transport*, vol. 505, chap. Computer-Aided Scheduling of Public Transport, pp. 37–52. Springer, Berlin Heidelberg.

Ceder, A. (2011) "Optimal multi-vehicle type transit timetabling and vehicle scheduling". *Procedia-Social and Behavioral Sciences*, vol. 20: 19–30.

Ceder, A. and Wilson, N. H. M. (1986) "Bus network design". *Transportation Research Part B: Methodological*, vol. 20(4): 331–344.

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

Chakroborty, P., Deb, K. and Sharma, R. K. (2001) "Optimal fleet size distribution and scheduling of transit systems using genetic algorithms". *Transportation Planning and Technology*, vol. 24(3): 209–226.

Chang, S.-C. and Chung, Y.-C. (2005) "From timetabling to train regulation–a new train operation model". *Information and Software Technology*, vol. 47(9): 575–585.

Cherkassky, B., Goldberg, A. and Radzik, T. (1996) "Shortest paths algorithms: Theory and experimental evaluation". *Mathematical Programming*, vol. 73: 129–174.

Chew, K., Pang, J., Liu, Q., Ou, J. and Teo, C. (2001) "An optimization based approach to the train operator scheduling problem at Singapore MRT". *Annals of Operations Research*, vol. 108(1): 111–118.

Christofides, N. and Korman, S. (1974) "A computational survey of methods for the set covering problem". *Management Science*, vol. 21(5): 591–599.

Church, R. L. and ReVelle, C. (1974) "The maximal covering location problem". *Papers of the Regional Science Association*, vol. 32: 101–118.

Colebrook, M. and Sicilia, J. (2007a) "A polynomial algorithm for the multicriteria cent-dian location problem". *European Journal of Operational Research*, vol. 179(3): 1008–1024.

Colebrook, M. and Sicilia, J. (2007b) "Undesirable facility location problems on multicriteria networks". *Computers & Operations Research*, vol. 34(5): 1491–1514.

Cominetti, R. and Correa, J. (1999) "The common-line problem in congested transit networks". *Transportation Science*, vol. 2001: 35–3.

Cominetti, R. and Correa, J. (2001) "Common-lines and passenger assignment in congested transit networks". *Transportation Science*, vol. 35(3): 250–267.

Conde, E. (2004) "An improved algorithm for selecting p items with uncertain returns according to the minmax-regret criterion". *Mathematical Programming*, vol. 100: 345–353.

Cordeau, J.-F., Laporte, G., Potvin, J.-Y. and Savelsbergh, M. W. (2007a) "Chapter 7: Transportation on Demand". In C. Barnhart and G. Laporte, eds., *Transportation*, vol. 14 of *Handbooks in Operations Research and Management Science*, pp. 429–466. Elsevier.

Cordeau, J.-F., Laporte, G., Savelsbergh, M. W. and Vigo, D. (2007b) “Chapter 6: Vehicle Routing”. In C. Barnhart and G. Laporte, eds., *Transportation*, vol. 14 of *Handbooks in Operations Research and Management Science*, pp. 367–428. Elsevier.

Croce, F. D., Paschos, V. T. and Tsoukias, A. (1999) “An improved general procedure for lexicographic bottleneck problems”. *Operations Research Letters*, vol. 24: 187–194.

Daskin, M. (1995) *Network and Discrete Location: Models, Algorithms, and Applications.* Wiley, New York.

Daskin, M. S. (2013) *Network and discrete location: models, algorithms, and applications, 2nd edition.* Wiley, Hoboken.

de Berg, M., Cheong, O., van Krefeld, M. and Overmars, M., eds. (2008) *Computational Geometry – Algorithms and Applications.* Springer-Verlag, third edition.

de Palma, A. and Lindsey, R. (2001) “Optimal timetables for public transportation”. *Transportation Research Part B: Methodological*, vol. 35(8): 789–813.

Desaulniers, G. and Hickman, M. D. (2007) “Chapter 2 public transit”. In C. Barnhart and G. Laporte, eds., *Transportation*, vol. 14 of *Handbooks in Operations Research and Management Science*, pp. 69–127. Elsevier.

Drezner, Z., ed. (1995) *Facility Location: A Survey of Applications and Methods.* Springer, New York.

Drezner, Z. and Hamacher, H., eds. (2002) *Facility Location: Applications and Theory.* Springer, New York.

Edelsbrunner, H. (1987) *Algorithms in Combinatorial Geometry.* Springer, New York, Berlin, Heidelberg.

Edmonds, J. (1965) “Paths, trees, and flowers”. *Canadian Journal of Mathematics*, vol. 17: 449–467.

Edmonds, J. (1970) *Submodular functions, matroids, and certain polyhedra*, pp. 69–87. Gordon and Breach, New York.

Edmonds, J. (1971) “Matroids and the greedy algorithm”. *Mathematical Programming*, vol. 1: 127–136.

Ehrgott, M. (2005) *Multicriteria Optimization.* Springer Berlin Heidelberg.

Ehrgott, M. and Gandibleux, X. (2000) “A survey and annotated bibliography of multiobjective combinatorial optimization”. *OR Spectrum*, vol. 22: 425–460.

Ehrgott, M. and Rau, A. (1999) “Bicriteria costs versus service analysis of a distribution network - a case study”. *Journal of Multi-Criteria Decision Analysis*, vol. 8: 256–267.

Eiselt, H. and Laporte, G. (1995) *Facility Loaction: A Survey of Applications and Methods*, chap. Objectives in location problems, pp. 151–180. Springer.

Espejo, I., Marin, A., Puerto, J. and Rodríguez-Chía, A. (2009) “A comparison of formulations and solution methods for the minimum-envy location problem”. *Computers & Operations Research*, vol. 36: 1966–1981.

Espejo, I., Marín, A. and Rodríguez-Chía, A. M. (2012) “Closest assignment constraints in discrete location problems”. *European Journal of Operational Research*, vol. 219(1): 49–58.

Fernández, E., Pozo, M. A. and Puerto, J. (2014) “Ordered weighted average combinatorial optimization: Formulations and their properties”. *Discrete Applied Mathematics*, vol. 169(0): 97–118.

Fernández, E., Pozo, M. A. P. and Puerto, J. (2013) “A modeling framework for ordered weighted average combinatorial optimization”. *arXiv:1306.1426.*

Fernández, E. and Puerto, J. (2003) “Multiobjective solution of the uncapacitated plant location problem”. *European Journal of Operational Research*, vol. 145: 509–529.

Fernández, E., Puerto, J. and Rodriguez-Chia, A. (2012) “On discrete optimization with ordering”. *Annals of Operations Research.* Doi: 10.1007/s10479-011-1044-7.

Fleurent, C. and Lessard, R. (2009) “Integrated timetabling and vehicle scheduling in practice”. Tech. rep., GIRO Inc. Montreal, Canada.

Fosgerau, M. (2009) “The marginal social cost of headway for a scheduled service”. *Transportation Research Part B: Methodological*, vol. 43(8–9): 813–820.

Galand, L. and Perny, P. (2007) "Search for choquet-optimal paths under uncertainty". In *Proceedings of the 23rd conference on uncertainty in artificial intelligence, AAAI Press*, pp. 125–132.

Galand, L. and Spanjaard, O. (2012) "Exact algorithms for OWA-optimization in multiobjective spanning tree problems". *Computers & Operations Research*, vol. 39: 1540–1554.

Gale, D. (1963) *Convexity*, vol. VII, chap. Neighborly and cyclic polytopes., pp. 225–232. Proceedings of the Symposia in Pure Mathematics. American Mathematics Society.

Galil, Z. and Schieber, B. (1998) "On finding most uniform spanning trees". *Discrete Applied Mathematics*, vol. 20: 173–175.

Gandibleux, X., Jaszkiewicz, A., Freville, A. and Slowinski, R. (2000) "Special issue in multiple objective metaheuristics". *Journal of Heuristics*, vol. 6: 291–431.

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

Garfinkel, R., Fernández, E. and Lowe, T. (2006) "The k-centrum shortest path problem". *TOP*, vol. 14: 279–292.

Gavish, B. (1983) "Formulations and algorithms for the capacitated minimal directed tree problem". *Journal of the ACM*, vol. 30: 118–132.

Gouveia, L. (1995) "Using the miller-tucker-zemlin constraints to formulate a minimal spanning tree problem with hop constraints". *Computers & Operations Research*, vol. 22(9): 959–970.

Grosfeld-Nir, A. and Bookbinder, J. (1995) "The planning of headways in urban public transit". *Annals of Operations Research*, vol. 60: 145–160.

Grötschel, M. and Holland, O. (1985) "Solving matching problems with linear programming". *Mathematical Programming*, vol. 33(3): 243–259.

Grzybowski, J., Nickel, S., Pallaschke, D. and Urbanski, R. (2011) "Ordered median functions and symmetries". *Optimization*, vol. 60(7): 801–811.

Guihaire, V. and Hao, J. (2008a) *Transit network re-timetabling and vehicle scheduling*, vol. 14, pp. 135–144. Springer, Berlin, Heidelberg.

Guihaire, V. and Hao, J.-K. (2008b) "Transit network design and scheduling: A global review". *Transportation Research Part A: Policy and Practice*, vol. 42(10): 1251–1273.

Guihaire, V. and Hao, J.-K. (2010) "Transit network timetabling and vehicle assignment for regulating authorities". *Computers & Industrial Engineering*, vol. 59(1): 16–23.

Guo, Z. and Wilson, N. H. (2011) "Assessing the cost of transfer inconvenience in public transport systems: A case study of the london underground". *Transportation Research Part A: Policy and Practice*, vol. 45(2): 91–104.

Gupta, M., Gupta, Y. and Bector, C. (1990) "Minimizing the flow-time variance in single-machine systems". *Journal of Operations Research Society*, vol. 41(8): 769–779.

Gupta, S. K. and Punnen, A. P. (1988) "Minimum deviation problems". *Operations Research Letters*, vol. 7(4): 201–204.

Hakimi, S. (1964) "Optimum locations of switching centers and the absolute centers and medians of a graph". *Operations Research*, vol. 12(3): 450–459.

Hakimi, S. (1965) "Optimum distribution of switching centers in a communications network and some related graph theoretic problems". *Operations Research*, vol. 13(3): 462–475.

Hamacher, H. and Nickel, S. (1996) "Multicriteria planar location problems". *European Journal of Operational Research*, vol. 94(1): 66–86.

Hamacher, H. and Ruhe, G. (1994) "On spanning tree problems with multiple objectives". *Annals of Operations Research*, vol. 52: 209–230.

Hamacher, H. W., Labbé, M. and Nickel, S. (1999) "Multicriteria network location problems with sum objectives". *Networks*, vol. 33(2): 79–92.

Hamacher, H. W., Labbé, M., Nickel, S. and Skriver, A. J. (2002) "Multicriteria semi-obnoxious network location problems (msnlp) with sum and center objectives". *Annals of Operations Research*, vol. 110: 33–53.

Hamdouch, Y., Ho, H., Sumalee, A. and Wang, G. (2011) "Schedule-based transit assignment model with vehicle capacity and seat availability". *Transportation Research Part B: Methodological*, vol. 45(10): 1805–1830.

Hamdouch, Y., Szeto, W. and Jiang, Y. (2014) "A new schedule-based transit assignment model with travel strategies and supply uncertainties". *Transportation Research Part B: Methodological*, vol. 67(0): 35–67.

Hansen, P. (1980) "Bicriterion path problems". In *Multiple criteria decision making theory and application*, vol. 177 of Lecture Notes in Econom. and Math. Systems, pp. 109–127.

Hansen, P. and Labbé, M. (1988) "Algorithms for voting and competitive location on a network". *Transportation Science*, vol. 22(4): 278–288.

Hansen, P., Labbé, M. and Thisse, J. (1991) "From the median to the generalized center." *RAIRO, Rech. Opér.*, vol. 25: 73–86.

Hershberger, J. (1989) "Finding the upper envelope of n line segments in o(n log n) time". *Information Processing Letters*, vol. 33(4): 169–174.

Higgins, A., Kozan, E. and Ferreira, L. (1996) "Optimal scheduling of trains on a single line track". *Transportation Research Part B: Methodological*, vol. 30(2): 147–161.

Hu, T. (1974) "Optimum communication spanning trees". *SIAM Journal on Computing*, vol. 3: 188–195.

Hurwicz, L. (1951) "Optimality criteria for decision making under ignorance". *Cowles Commission Discussion Paper, Chicago*, p. 370.

Ibarra-Rojas, O. J., Giesen, R. and Rios-Solis, Y. A. (2014) "An integrated approach for timetabling and vehicle scheduling problems to analyze the trade-off between level of service and operating costs of transit networks". *Transportation Research Part B: Methodological*, vol. 70(0): 35–46.

Johnson, M. (2001) "A spatial decision support system prototype for housing mobility program planning". *Journal of Geographical Systems*, vol. 3: 49–67.

Kalcsics, J. (2011) "The multi-facility median problem with pos/neg weights on general graphs". *Computers & Operations Research*, vol. 38: 674–682.

Kalcsics, J., Nickel, S., Puerto, J. and Rodríguez-Chía, A. M. (2015) "Several 2-facility location problems on networks with equity objectives". *Networks*, vol. 65(1): 1–9.

Kalcsics, J., Nickel, S., Puerto, J. and Tamir., A. (2002) "Algorithmic results for ordered median problems defined on networks and the plane". *Operations Research Letters*, vol. 30: 149–158.

Kasperski, A. and Zielinski, P. (2013) "Combinatorial optimization problems with uncertain costs and the OWA criterion". *arXiv:1305.5339*.

Kelly, L. M. (1944) "Covering problems". *National Mathematics Magazine*, vol. 19: 123–130.

Kolokolov, A. and Zaozerskaya, L. (2013) "Solving a bicriteria problem of optimal service centers location". *Journal of Mathematical Modelling and Algorithms in Operations Research*, vol. 12(2): 105–116.

Kruskal, J. (1956) "On the shortest spanning subtree of a graph and the traveling salesman problem". *Proceedings of the American Mathematical Society*, vol. 7: 48–50.

Kumazawa, K., Hara, K. and Koseki, T. (2010) *A novel train rescheduling algorithm for correcting disrupted train operations in a dense urban environment.* Wit Press/Computational Mechanics.

Landete, M. and Marín, A. (2014) "Looking for edge-equitable spanning trees". *Computers & Operations Research*, vol. 41(0): 44–52.

Laporte, G. (1992) "The traveling salesman problem: An overview of exact and approximate algorithms". *European Journal of Operational Research*, vol. 59(2): 231–247.

Laporte, G., Mesa, J. and Ortega, F. (1994) “Assessing topological configurations for rapid transit networks”. *Stud Locat Anal*, vol. 7: 105–121.

Laporte, G., Mesa, J. and Ortega, F. (1997) “Assesing the efficiency of rapid transit configurations”. *Top*, vol. 5(1): 95–104.

Laporte, G., Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2009) “Locating a metro line in a historical city centre: application to sevilla”. *Journal of the Operational Research Society*, vol. 60(5): 1462–1466.

Laporte, G., Nickel, S. and Saldanha da Gama, F., eds. (2015) *Location Science*. Springer International Publishing Switzerland.

Larson, R. and Odoni, A. (1981) *Urban operations research*. Prentice-Hall: New York.

Lawler, E. L. (1966) “Covering problems: Duality relations and a new method of solution”. *SIAM Journal on Applied Mathematics*, vol. 14: 1115–1132.

Lawler, E. L. (1972) “A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem”. *Management Science*, vol. 18(7): 401–405.

Lei, T. L. and Church, R. L. (2012) “Vector assignment ordered median problem: A unified median problem”. *International Regional Science Review*. Doi: 10.1177/0160017612450710.

Lesca, J., Minoux, M. and Perny, P. (2013) “Compact versus noncompact LP formulations for minimizing convex choquet integrals”. *Discrete Applied Mathematics*, vol. 161: 184–199.

Liebchen, C., Schachtebeck, M., Schöbel, A., Stiller, S. and Prigge, A. (2010) “Computing delay resistant railway timetables”. *Computers & Operations Research*, vol. 37(5): 857–868.

Liu, Z.-G. and Shen, J.-S. (2007) “Regional bus operation bi-level programming model integrating timetabling and vehicle scheduling”. *Systems Engineering–Theory & Practice*, vol. 27(11): 135–141.

Lozano, A. J., Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2008) “Assessing the robustness in topological configurations of rapid transit systems”. In *Proceedings of the XI ISOLDE (International Symposium on Locational Decisions)*, vol. 1, pp. 91–91. University of California in Santa Barbara.

López de los Mozos, M., Mesa, J. and Puerto, J. (2008) “A generalized model of equality measures in network”. *Computers & Operations Research*, vol. 35: 651–660.

Lusby, R., Larsen, J., Ehrgott, M. and Ryan, D. (2011) “Railway track allocation: models and methods”. *OR Spectrum*, vol. 33 (4): 843–883.

Magnanti, T. and Wong, R. (1984) “Network design and transportation planning: Models and algorithms.” *Transportation Science*, vol. 18(1): 1–55.

Magnanti, T. L. and Wolsey, L. A. (1995) “Optimal trees”. In C. M. M.O. Ball, T.L. Magnanti and G. Nemhauser, eds., *Network Models, Chapter 9*, vol. 7 of *Handbooks in Operations Research and Management Science*, pp. 503 – 615. Elsevier.

Marín, A., Nickel, S., Puerto, J. and Velten, S. (2009) “A flexible model and efficient solution strategies for discrete location problems”. *Discrete Applied Mathematics*, vol. 157(5): 1128–1145.

Martello, S., Pulleyblank, W., Toth, P. and de Werra, D. (1984) “Balanced optimization problems”. *Operations Research Letters*, vol. 3(5): 275–278.

Martin, R. (1991) “Using separation algorithms to generate mixed integer model reformulations”. *Operations Research Letters*, vol. 10(3): 119–128.

Matisziw, T. C. and Murray, A. T. (2009) “Modeling s-t path availability to support disaster vulnerability assessment of network infrastructure”. *Computers & Operations Research*, vol. 36(1): 16–26.

Mesa, J., Puerto, J. and Tamir, A. (2003) “Improved algorithm for several network location problems with equality measures”. *Discrete Applied Mathematics*, vol. 130: 437–448.

Mesa, J. A., Ortega, F., Ángel Pozo, M. and Puerto, J. (2014a) "Rescheduling railway timetables in presence of passenger transfers between lines within a transportation network". In J. F. Sousa and R. Rossi, eds., *Computer-based Modelling and Optimization in Transportation*, vol. 262 of *Advances in Intelligent Systems and Computing*, pp. 347–360. Springer International Publishing Switzerland.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2008) "Robustez ante incidencias en el diseño topológico de redes metropolitanas de transporte público". In *Actas del VIII CIT (Congreso de Ingeniería del Transporte)*, vol. 1, pp. 1–8. Demarcación del Colegio de Ingenieros de Caminos, Canales y Puertos.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2009a) "Determining railway timetables according to users' desires and fleet size constraints". In *Proceedings of the XXIII EURO (European Conference on Operational Research)*, vol. 1, pp. 151–151.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2009b) "Effective allocation of fleet frequencies by reducing intermediate stops and short turning in transit systems". *Lecture Notes in Computer Science*, vol. 5868(8): 293–309.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2013) "A geometric model for an effective rescheduling after reducing service in public transportation systems". *Computers & Operations Research*, vol. 40(3): 737–746.

Mesa, J. A., Ortega, F. A. and Pozo, M. A. (2014b) "Locating optimal timetables and vehicle schedules in a transit line". *Annals of Operations Research*, vol. 222: 439–455.

Miller, C. E., Tucker, A. W. and Zemlin, R. A. (1960) "Integer programming formulation of traveling salesman problems". *Journal of the ACM*, vol. 7(4): 326–329.

Minoux, M. (1989) "Network synthesis and optimum network design problems: models, solution methods and applications". *Networks*, vol. 19: 313–360.

Mirchandani, P. and Francis, R. (1990) *Discrete Location Theory*. John Wiley and Sons, New York.

Murray, A. T. (2003) "A coverage model for improving public transit system accessibility and expanding access". *Annals of Operations Research*, vol. 123: 143–156.

Nakamura, T., Hirai, C. and Nishioka, Y. (2011) "A practical train rescheduling algorithm using three predetermined factors". In *Proceedings of the 4th International Seminar on Railway Operations Modelling and Analysis*.

Nickel, S. and Puerto, J. (2005) *Location Theory: A Unified Approach*. Springer-Verlag, Heidelberg, Germany.

Nickel, S., Puerto, J. and Rodríguez-Chía, A. M. (2005) "MCDM location problems". In J. A. Figueira, S. Greco and M. Ehrogott, eds., *Multiple Criteria Decision Analysis: State of the Art Surveys*, vol. 78 of *International Series in Operations Research & Management Science*, pp. 761–787. Springer, New York.

Nickel, S., Puerto, J. and Rodriguez-Chia, A. (2015) "Location problems with multiple criteria". In G. Laporte, S. Nickel and F. Saldanha da Gama, eds., *Location Science*, chap. 9, pp. 205–247. Springer International Publishing Switzerland.

Nuzzolo, A., Crisalli, U. and Rosati, L. (2012) "A schedule-based assignment model with explicit capacity constraints for congested transit networks". *Transportation Research Part C: Emerging Technologies*, vol. 20(1): 16–33. Special issue on Optimization in Public Transport+ISTT2011 Special issue on Optimization in Public Transport+International Symposium on Transportation and Traffic Theory (ISTTT), Berkeley, California, July 18–20, 2011.

Ogryczak, W. and Olender, P. (2012) "On MILP models for the OWA optimization". *Journal of Telecommunications and Information Technology*, vol. 2/2012: 5–12.

Ogryczak, W. and Sliwinski, T. (2003) "On solving linear programs with the ordered weighted averaging objective". *European Journal of Operational Research*, vol. 148(1): 80–91.

Ogryczak, W. and Tamir, A. (2003) "Minimizing the sum of the k largest functions in linear time". *Information Processing Letters*, vol. 85 (3): 117–122.

Oliveira, E. and Smith, B. M. (2000) "A job-shop scheduling model for the single-track railway scheduling problem". Tech. rep., 2000.21, University of Leeds, Leeds, UK.

Pascoal, M., Captivo, M. E. and Clímaco, J. (2003) “A note on a new variant of Murty’s ranking assignments algorithm”. *4OR*, vol. 1(3): 243–255.

Pel, A. J., Bel, N. H. and Pieters, M. (2014) “Including passengers’ response to crowding in the dutch national train passenger assignment model”. *Transportation Research Part A: Policy and Practice*, vol. 66(0): 111–126.

Perny, P. and Spanjaard, O. (2003) “An axiomatic approach to robustness in search problems with multiple scenarios.” In *Proceedings of the 19th conference on uncertainty in artificial intelligence*, pp. 469–476.

Petersen, H., Larsen, A., Madsen, O., Petersen, B. and Ropke, S. (2013) “The simultaneous vehicle scheduling and passenger service problem”. *Transportation Science*, vol. 47: 603–616.

Poon, M. H., Wong, S. C. and Tong, C. O. (2004) “A dynamic schedule-based model for congested transit networks”. *Transportation Research Part B: Methodological*, vol. 38(4): 343–368.

Prim, R. (1957) “Shortest connection networks and some generalizations”. *Bell System Technical Journal*, vol. 36: 1389–1401.

Puerto, J. (1996) “Lecturas en teoría de localización”. Tech. rep., Universidad de Sevilla. Secretariado de Publicaciones.

Puerto, J. (2008) “A new formulation of the capacitated discrete median problem with $\{0,1\}$-assignment”. In J. Kalcsics and S. Nickel, eds., *Operations Research Proceedings 2007*, chap. , pp. 165–170. Springer, Heidelberg.

Puerto, J. and Fernández, F. (1999) “Multicriteria minisum facility location problem”. *Journal of Multi-Criteria Decision Analysis*, vol. 8: 268–280.

Puerto, J. and Rodríguez-Chía, A. (2003) “Robust positioning of service units”. *Stochastic Models*, vol. 19(1): 125–147.

Puerto, J. and Tamir, A. (2005) “Locating tree-shaped facilities using the ordered median objective”. *Mathematical Programming*, vol. 102(2): 313–338.

Punnen, A. (1992) “Traveling salesman problem under categorization”. *Operations Research Letters*, vol. 12(2): 89–95.

Punnen, A., Nair, K. and Aneja, Y. (1995) “Generalized bottleneck problems”. *Optimization*, vol. 35(2): 159–169.

Punnen, A. P. and Aneja, Y. P. (1997) “Minimum dispersion problems”. *Discrete Applied Mathematics*, vol. 75(1): 93–102.

Quak, C. (2003) *Bus line planning.* Ph.D. thesis, Delft University of Technology. The Netherlands.

Ramaswamy, R., Orlin, J. B. and Chakravarti, N. (2005) “Sensitivity analysis for shortest path problems and maximum capacity path problems in undirected graphs”. *Mathematical Programming*, vol. 102(2): 355–369.

Ross, G. and Soland, R. (1980) “A multicriteria approach to the location of public facilities”. *European Journal of Operational Research*, vol. 4: 307–321.

Sato, K., Tamura, K. and Tomii, N. (2013) “A mip-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers”. *Journal of Rail Transport Planning & Management*, vol. 3(3): 38–53.

Schöbel, A. and Scholl, S. (2006) “Line planning with minimal traveling time”. In *Dagstuhl Seminar Proceedings*, p. .

Schmöcker, J.-D., Bell, M. G. and Kurauchi, F. (2008) “A quasi-dynamic capacity constrained frequency-based transit assignment model”. *Transportation Research Part B: Methodological*, vol. 42(10): 925–945.

Schmeidler, D. (1986) “Integral representation without additivity”. *Prooceedings of the American Mathematical Society*, vol. 97(2): 255–261.

Schmidt, M. and Schöbel, A. (2015a) “The complexity of integrating passenger routing decisions in public transportation models”. *Networks*, vol. 65(3): 228–243.

Schmidt, M. and Schöbel, A. (2015b) “Timetabling with passenger routing”. *OR Spectrum*, vol. 37(1): 75–97.

Schöbel, A. (2005) "Locating stops along bus or railway lines–a bicriteria problem". *Annals of Operations Research*, vol. 136(1): 211–227.

Schrijver, A. (1983) *Min-max results in combinatorial optimization*, pp. 439–500. Springer, Berlin.

Sharir, M. (1994) "Almost tight upper bounds for lower envelopes in higher dimensions". *Discrete & Computational Geometry*, vol. 12: 327–345.

Shier, R. D. (1979) "On algorithms from finding the k shortest paths in a network". *Networks*, vol. 9: 195–214.

Siebert, M. and Goerigk, M. (2013) "An experimental comparison of periodic timetabling models". *Computers & Operations Research*, vol. 40(10): 2251–2259.

Skriver, A. J., Andersen, K. A. and Holmberg, K. (2004) "Bicriteria network location (bnl) problems with criteria dependent lengths and minisum objectives." *European Journal of Operational Research*, vol. 156: 541–549.

Slater, P. (1978a) "Centers to centroids in graphs". *Journal of Graph Theory*, vol. 2(3): 209–222.

Slater, P. (1978b) "Structure of the k-centra in a tree". In *Proceedings of the Ninth Southeastern Conference on Combinatorics, Graph Theory, and Computing*, vol. XXI, pp. 663–670.

Small, K. A. (1982) "The scheduling of consumer activities: Work trips". *The American Economic Review*, vol. 72(3): 467–479.

Sourd, F. and Spanjaard, O. (2008) "A multi-objective branch-and-bound framework. application to the bi-objective spanning tree problem". *INFORMS Journal of Computing*, vol. 20 (3): 472–484.

Spiess, H. and Florian, M. (1989) "Optimal strategies: A new assignment model for transit networks". *Transportation Research Part B: Methodological*, vol. 23(2): 83–102.

Steiner, S. and Radzik, T. (2008) "Computing all efficient solutions of the biobjective minimum spanning tree problem". *Computers & Operations Research*, vol. 35 (1): 198–211.

Stern, R. (1996) "Passenger transfer system review. synthesis of transit practice". *Transportation Research Board*, vol. 19.

Tamir, A. (2000) "The k-centrum multi-facility location problem". *Dis*, vol. 109: 292–307.

Tamir, A., Puerto, J. and Perez-Brito, D. (2002) "The centdian subtree on tree networks". *Discrete Applied Mathematics*, vol. 118: 263–278.

Tansel, B. C., Francis, R. L. and Lowe, T. J. (1982) "Biobjective multifacility minimax location problem on a tree network." *Transportation Science*, vol. 16(4): 407–429.

Törnquist, J. (2007) "Railway traffic disturbance management–an experimental analysis of disturbance complexity, management objectives and limitations in planning horizon". *Transportation Research Part A: Policy and Practice*, vol. 41(3): 249–266.

Ulungu, E. and Teghem, J. (1994) "Multi-objective combinatorial optimization problems: a survey". *Multi-Criteria Decision Analysis*, vol. 3: 83–104.

Van de Velde, D. (1999) "Organisational forms and entrepreneurship in public transport–part 1: Classifying organisational forms". *Transport Policy*, vol. 6: 147–157.

van den Heuvel, A., van den Akker, J. and van Kooten, M. (2008) "Integrating timetabling and vehicle scheduling in public bus transportation". Tech. rep., UU-CS-2008-003. Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands.

van Exel, N. A. and Rietveld, P. (2001) "Public transport strikes and traveller behaviour". *Transport Policy*, vol. 8(4): 237–246.

Vieira, G., Herrmann, J. and Lin, E. (2003) "Rescheduling manufacturing systems: A framework of startegies, policies, and methods". *Journal of Scheduling*, vol. 6: 39–62.

Wagner, J. L. and Falkson, L. M. (1975) "The optimal nodal location of public facilities with price-sensitive demand". *Geographical Analysis*, vol. 7(1): 69–83.

Warburton, A. (1983) “Quasiconcave vector maximization: connectedness of the sets of pareto-optimal and weak pareto-optimal alternatives.” *Journal of Optimization Theory and Applications*, vol. 40: 537–557.

Weaver, J. R. and Church, R. L. (1985) “A median location model with non-closest facility service”. *Transportation Science*, vol. 19: 58–74.

Wendell, R. and Hurter, A. (1973) “Location theory, dominance and convexity”. *Operations Research*, vol. 21: 314–320.

Wirasinghe, S. (2003) *Advanced Modeling for Transit Operations and Service Planning*, chap. Initial planning for urban transit systems., pp. 1–29. Elsevier Science.

Wren, A. and Rousseau, J. (1995) *Computer-Aided Transit Scheduling*, chap. Bus driver scheduling–an overview, pp. 173–187. .

Yager, R. (1988) “On ordered weighted averaging aggregation operators in multicriteria decision making”. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18: 183–190.

Yager, R. and Kacprzyk, J. (1997) *The Ordered Weighted Averaging Operators: Theory and Applications*. Kluwer, Norwell, MA.

Yen, J. Y. (1971) “Finding the K shortest loopless paths in a network”. *Management Science*, vol. 17: 712–716.

Yu, B., Yang, Z., Cheng, C. and Liu, C. (2005) “Optimizing bus transit network with parallel ant colony algorithm”. In *Proceedings of the Eastern Asia Society for Transportation Studies*, vol. 5, pp. 374–389.

Yu, G. (1998) “Min-max optimization of several classical discrete optimization problems.” *Journal of Optimization Theory and Applications*, vol. 98(1): 221–42.

Zhang, F. and Melachrinoudis, E. (2001) “The maximin-maxisum network location problem”. *Computational Optimization and Applications*, vol. 19(2): 209–234.

Zhou, X. and Zhong, M. (2005) “Bicriteria train scheduling for high-speed passenger railroad planning applications”. *European Journal of Operational Research*, vol. 167(3): 752–771.

Zhou, X. and Zhong, M. (2007) “Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds”. *Transportation Research Part B: Methodological*, vol. 41(3): 320–341.