Ph.d. Thesis

Università degli Studi di Napoli "Federico II"

Dipartimento di Informatica e Sistemistica

Dottorato di Ricerca in
Ingegneria Informatica ed Automatica

Location-Routing models and methods for Freight Distribution and Infomobility in City Logistics

---

Claudio Sterle

Tutor: Ch.mo Prof. Antonio Sforza

Co-Tutor: Ch.mo Prof. Teodor Gabriel Crainic

Coordinatore del Corso di Dottorato: Ch.mo Prof. Francesco Garofalo

2009

Acknowledgments

I want to express my sincere gratitude to Professor Antonio Sforza for his valuable guidance and support during my time as a Ph.D. student. He provides me everything I needed to succeed in the research. I also thank him for proof-reading this manuscript.

I have to give a grateful thought also to Professor Teodor Gabriel Crainic, who welcomed me at CIRRELT research center in Montreal and gave me the opportunity to live there and work with him.

I think that the results here collected are the fruits of a team-working.

Finally, I am forever indebted to my parents and to Michela, for their support, understanding, endless patience and encouragement when it was most required.

Claudio Sterle

Contents

Acknowledgments **iii**

Introduction **1**

I City Logistics, Freight Distribution and Infomobility **7**

1 City Logistics: definition and strategies **9**
1.1 Congestion of urban areas 9
1.2 City Logistics 12
1.3 City Logistics measures and strategies 14

2 Freight Distribution and Infomobility **19**
2.1 Freight distribution problem 19
2.1.1 Decisional levels and stakeholders 23
2.1.2 Single-echelon freight distribution system 25
2.1.3 Two-echelon freight distribution system 26
2.1.4 Optimization for freight distribution 29
2.2 Infomobility 29
2.2.1 Infomobility and information 35
2.2.2 Infomobility systems 36
2.2.3 Optimization for infomobility 38

II Two-echelon location-routing problem 2E-LRP **41**

3 Location-routing problem definition **43**

3.1 FLP and LRP . . . 43
3.2 Generalized LRP . . . 45
3.3 Literature review . . . 48
3.4 A two echelon location-routing problem (2E-LRP) for freight distribution . . . 54

4 Models for the 2E-LRP 59
4.1 2E-LRP Setting . . . 59
4.2 A three-index 2E-LRP formulation . . . 60
4.3 A two-index 2E-LRP formulation . . . 65
4.4 A one-index 2E-LRP formulation . . . 69
4.5 Assignment-based 2E-LRP formulation . . . 72
4.6 2E-LRP models computational results . . . 77

5 Tabu Search heuristic for 2E-LRP 83
5.1 Solution approaches . . . 83
5.2 Introduction to TS . . . 85
5.3 A tabu search heuristic for 2E-LRP . . . 87
5.4 First feasible solution and evaluation . . . 89
5.4.1 Estimated and actual cost of a solution . . . 92
5.5 Solution neighborhood definition . . . 94
5.6 Location moves . . . 94
5.6.1 Swap moves . . . 94
5.6.2 Add move . . . 97
5.7 Routing moves . . . 97
5.7.1 Multi-stop routes definition and improvement . . . 98
5.7.2 Intra-routes improvements for a single facility . . . 100
5.7.3 Intra-routes improvements for multiple facilities . . . 102
5.8 Combining sub-problems . . . 104
5.8.1 Combining sub-problems of a single echelon . . . 104
5.8.2 Combining sub-problems of the two echelons . . . 104
5.9 Diversification criteria . . . 108

6 Computational results of TS for 2E-LRP 109
6.1 TS results and settings . . . 109
6.2 Comparisons of TS with exact methods . . . 117

III Flow intercepting facility location (FIFLP): problems, models and heuristics 125

7 Problem definition and related models 127
7.1 Introduction to FIFLP . . . 127
7.2 Literature review . . . 128
7.3 Five key issues in problem definition . . . 132
7.4 FIFLP formulations . . . 135
7.5 Flow oriented problems . . . 135
7.5.1 Problem *P1*: maximization of the intercepted flow 136
7.5.2 Problem *P2*: minimization of the number of flow intercepting facilities . . . 138
7.6 Gain oriented problems . . . 139
7.6.1 Problem *P3*: maximization of the achievable gain 139
7.6.2 An empirical expression for the a_{pi} coefficients . . 140
7.6.3 Problem *P4*: minimization of the number of facilities for gain maximization . . . 143
7.7 FIFLP extensions . . . 143
7.8 Mobile facility location problem . . . 146
7.8.1 Iterative approach . . . 146
7.8.2 Dynamic approach . . . 148

8 Heuristic approaches for FIFLP 151
8.1 Two greedy heuristics for problem *P1* . . . 151
8.2 An ascent heuristic for *P2* . . . 153
8.3 Heuristic for multiple FIFLP . . . 155
8.4 A greedy heuristic for *P3* . . . 156
8.5 An ascent search heuristic for *P3* . . . 156
8.6 A tabu search heuristic for *P3* . . . 158
8.7 A greedy heuristic for *P4* . . . 159

9 Computational results for FIFLP 161
9.1 Experimental tests on grid and random networks . . . 161
9.2 Graphical representation of the experimental results on a small network . . . 162
9.2.1 Intercepted flows vs. number of facilities for *P1* and *P2* . . . 162
9.2.2 Achieved gain vs. number of facilities for problem *P3* and *P4* . . . 165

9.3 Results of experimental tests 168
9.4 Sensitivity analysis and notations 183

Conclusion **189**

Appendix A **191**

Introduction

Traffic congestion is one of the main problems which affect urban areas. Great interest is devoted to this theme to reduce pollution and improve quality of life. In particular focusing on the effects of freight transportation the concept of *City Logistics* has been developed.

City Logistics is devoted to the management of the urban mobility, based on developing and performing strategical and tactical operations aimed at guarantee people and freight mobility in an effective way, in terms of social and environmental costs. Tactical operations are based on the regulation of the access of vehicles in the city center. Strategical operations are based on the design of freight distribution system. This requires the usage of logistic platforms located on the outskirt of the city center and devoted to freight collection and distribution.

This thesis concerns the application of Operation Research location and location-routing models and methods to two City Logistics problems.

The first is a design problem for a two-echelon freight distribution system. The aim is to define the structure of a system optimizing the location and the number of two different kinds of facilities, the size of two different vehicle fleets (urban trucks and city freighters) and the related routes on each echelon. The problem has been modeled as a two-echelon (multilevel) location-routing problem (2E-LRP). This problem is NP-hard since it arises from the combination of two NP-hard problems, facility location (FLP) and vehicle routing (VRP). At the best of our knowledge, multi-level location-routing problems have not yet been addressed either with exact or heuristic methods.

The second problem concerns the location of flow intercepting facilities. Differently to what happens in classical location problems, in this case facilities do not generate and/or attract flows, but they intercept flows traveling on the network. These facilities can be used by the flow units of the network or proposed to/imposed on them along their preplanned path from an origin to a destination. The aim is to define the locations of the facilities which optimize a performance criterion related to the flow values on each path. This is a path-covering problem which finds many applications in City Logistics, in particular the location of traffic monitoring and control facilities (i.e. variable message signs, sensors, inspection stations,etc.). Applications can be found also in the field of communication networks to locate monitoring devices (monitors or probes) which, placed inside the routers or deployed as a standalone box on the links of a communication network, summarize and record information about traffic flows, in order to prevent attack to network infrastructures.

The thesis is structured in three parts. *Part 1* is composed of two chapters. In *Chapter 1* the congestion problem for the urban areas is presented, highlighting the related negative effects from the social, economical and environmental point of view. Then the discussion is focused on the definition of *City Logistics*, with a presentation of its main targets, strategies and results in several national and international experiences. In *Chapter 2*, two issues of the *City Logistics* are presented more in detail: freight distribution and Infomobility. Concerning the first point, a description of the decisional levels to take into account in the design of a freight distribution system is provided. Then the inefficiencies of single-echelon freight distribution systems are discussed, to conclude with the idea of a multi-level freight-distribution system. A brief description of the related optimization problems is provided. Concerning the second point, a presentation of the main Infomobility concepts is provided, highlighting strategies and results in terms of safety, efficiency and environment. Then the discussion is focused on the information, which is the key element of the Infomobility, providing a definition and several classifications for it. Finally a brief presentation of the infomobility facilities and of the deriving location problems is provided.
Part 2 is composed of four chapters. It concerns the location-routing problems arising in the design of a multi-echelon freight distribution system. In *Chapter 3*, generalized location-routing problems (LRP) are presented, providing a definition and a classification for them. Then a wide literature review is proposed. Finally LRP is extended to the two-echelon case, with a discussion about the basic assumptions for their

application in freight distribution system design problems. In *Chapter 4*, four mixed-integer formulations for the two-echelon location-routing problems are provided. The first three formulations are directly derived from the classical formulations for the VRP, whereas the last one, is an adaptation of a formulation proposed for the multi-depot VRP. The chapter concludes with a presentation of several results obtained on small and medium instances for two of the proposed models. The computational results, obtained with a commercial solver, show that the computation time significantly increase with the size of the problem and therefore a heuristic approach is required to tackle large size real instances. Hence in *Chapter 5* a *Tabu Search* heuristic approach for the two-echelon location-routing problem is presented. The chapter starts with a discussion about the different solution approaches present in literature, based on problem decomposition. Then it focuses on the proposed method, with a presentation of the different steps of a tabu search heuristic: definition of an initial solution, definition of the neighborhood of a solution and related tabu settings, stopping and diversification criteria. In *Chapter 6* results of the proposed Tabu Search on three sets of small, medium and large instances are presented. The three sets differ for the spatial distribution of the secondary facilities. Each test set has been solved with different settings of the tabu search parameters. Results have been compared with the solutions provided by a commercial solver for the whole problem. The obtained results show that the proposed Tabu Search is able to find good solutions, if compared with available bounds, with limited computation time.
Part 3 is composed of two chapters. It concerns the usage of flow intercepting facility location models and methods for Infomobility services. In *Chapter 7* a presentation of flow interception problem is provided, with a focus on the basic issues for the problem definition. Four fixed flow intercepting facility location problems are treated. Each of them has been formulated as a mixed-integer model, which differs for the functions defined on the path to intercept. The chapter concludes with a presentation of several modifications of the proposed models and with an adaptation of them to the mobile facility case. These problems are NP-hard and therefore heuristic approaches are required for large size instances. Therefore in *Chapter 8* several greedy, ascent and meta heuristics for the four problems are presented. Finally in *Chapter 9* proposed models and methods have been experienced on test networks of varying

dimension and topology (mesh and random), comparing the obtained results in terms of quality of solution and computation times. The chapter concludes with a sensitivity analysis in function of several settings and characteristics of the problems under investigation (for example range of the flow values, number of paths and facilities), in order to verify the effect of these parameters. From the performed experimentation we can adfirm that heuristics return good solutions, very close to the optimum, even if in some cases (for networks with less than 200 nodes and for large values of facilities to locate) they require computation times not far from those required by the mathematical models. Therefore we have to carefully consider the settings of the problem and the trade off between quality of solution and computation times.

Part I

City Logistics, Freight Distribution and Infomobility

Chapter 1

City Logistics: definition and strategies

In this chapter traffic congestion problems for the urban areas will be presented, highlighting the related negative effects from the social, economical and environmental point of view. Then the discussion is focused on the definition of *City Logistics*, with a presentation of its main components, targets, strategies and results.

1.1 Congestion of urban areas

The congestion of the urban and regional areas is a relevant problem and the deriving emergencies call the local and national governments to adopt *logistic* measures to reduce the negative effects, improving the *mobility* within the areas under investigation.

The globalization has significantly changed the way of doing business. For this reason, a territory, which does not offer an efficient logistic service to satisfy the *transportation* demand, will have many problems in its economic, social and environmental development.

In the last fifty years great interest has been addressed to the development and consolidation of studies concerning the mobility system on urban and regional scale, focusing on its two main components: freight transportation and people transportation (individual and collective). At first people and freight mobility management was treated by empiric and experimental approaches, whereas, from the '90s, the complexity of the problems arising in this context and of the possible solutions has brought to the usage of methodologically advanced approaches (mathematical models, simulation methods, support decision systems, etc.).

Based on these methodological approaches, in the last twenty years, urban traffic management centers have been realized in medium and large cities. These centers are basically devoted to functions which are typical of the traffic management: video-surveillance, detection and/or monitoring of the traffic flows, control of the pedestrian areas and of the restricted traffic zones, traffic light regulation and control of the infractions.

This solution, which at the beginning seems very promising, is showing some deficiencies, due to present characteristics of the mobility system. In fact, nowadays, the mobility system on urban and regional scale are characterized by an increasing number of components, concerning both supply and demand of transportation. More precisely demand of transportation is significantly increased in the years on one side for the increase of movements related to the freight transportation and on the other side for the wide variety of people transportation demand.

Therefore the state of congestion of urban and regional network is not only determined by the traffic volumes due to the classical transportation demand (job, study, services, free time, etc.), but it is also due to that components of the transportation demand which are related to the freight movements (classified for vehicle dimension, time windows, requirements of charge/discharge areas) and to the development of special urban services (garbage collection, school transportation, ambulances, civil protection, tourism, etc.).

All these components use the same network infrastructures, and in several areas or time windows, they can assume a relevant role in increasing the level of congestions and in modifying the normal traffic conditions.

Moreover if freight and people movements are performed using not environmental friendly vehicles and are not well organized, then this means pollution and negative externalities for the urban areas. The negative effects can be identified not only in the congestion and air pollution, but also in the noise pollution, energy and work time consumption, low level of safety on the roads, damages and deterioration of infrastructures and of the historical centers. One of the main cause of this situation can be identified in the fact that the most part of the movements taking place in a urban area are performed by the most polluting mean of transportation, i.e. road transportation.

Hence it is clear that a good management and organization of the mobility demand can contribute to the reduction of the congestion phenomenon and to the decrease of the deriving externalities.

A proof of the increasing interest in these problems can be easily found in several European and national guidelines proposed for the improvement of the mobility system. Here a fast overview of the disposals at national and European scale is presented:

- December 1992, "Libro bianco dei trasporti-Lo sviluppo futuro della politica comune dei trasporti", published by the European Commission. The key point of this document is the opening of the market. This target has been reached in the following ten years with the only exception of the railway transportation.

- March 1998, the decree "Mobilitá sostenibile nelle aree urbane", emanated by the Minister of the Environment and Transportation, proposes to find alternative solutions in order to improve the movements of the residents from the houses to the place of work.

- September 2001, "Libro bianco dei trasporti", proposed by the European Commission. It established common strategies for the management of the transportation system: balancing the usage of the different means of transportation, sustaining the intermodality, decreasing of the congestion levels, improving the mobility but taking into account the safety and the quality of the offered services in the urban areas.

- May 2005, "Patto per la logistica" emanated by the Minister of the Infrastructure and Transportation. This deal concerns the definition of city logistic measures to decrease the effect on the environment. The main measures will be focused in the optimization of the distribution, with particular reference to: strengthening of the city infrastructures; fleet optmization and regulations.

- January 2006, "Piano per la logistica" emanated by the Minister of the Infrastructure and Transportation. This deal following the previous disposal, concerns the definition of the city logistic measures to put in act on four main aspects: infrastructure, safety, intermodality and regulation of the commercial transactions.

From the previous discussion it is easy to understand that the dimensions of the problem are rapidly increasing. Moreover it is not going to diminishes in the future, since in the last fifty years a world-wide urbanization phenomenon is taking place, emptying countryside and small towns and making large cities even larger. To better explain, concerning the OECD countries members [81], the urban population was 50% of the total population in 1950, was 77% in 2000, and should reach the 85% mark by 2020. For what concerns instead the italian case, it has been observed that the 53% of the whole population lives in the 14 bigger metropolitan areas [100].

This situation has brought to the definition of new problems in the context of the urban mobility and consequently of the logistics. On the other side this has brought to new challenges and opportunities for the application of OR methods.

1.2 City Logistics

City Logistics is defined as the "process for the optimization of all the transportation activities which take place in a urban area, considering its effects in terms of impacts on the traffic, congestion, energy consumption and on the economic life of the area" (Taniguchi et al. [93] and [92]). Therefore City Logistics is planning, implementing and management of the physical and informative flows in a urban area, in order to have a good urban mobility system, served by an effective and efficient transportation system.

The main City Logistics measures can be classified as follows:

1. Introduction of Intelligent Transportation System and Telematic infrastructures.

2. Management of people and freight transportation in urban areas.

3. Traffic control and management, with particular reference to the environmental problems.

4. Multi-modality, i.e. intelligent usage of different sustainable modes of transportation.

5. Modification of the behavior in the usage of the network.

6. Cooperation among all the actors of the mobility system.

So it can be adfirmed that City Logistics is aimed at the management of the mobility demand, developing and implementing strategies with the purpose of determining an efficient mobility for people and freights in order to obtain good results in terms of social, economic, environmental and urban benefits ([92]), [73], [100]). More precisely:

- ***Social:***

- create new sources of employment;
- improve the working condition for the interested people;
- improve the life quality in the urban areas;
- increase the safety of the 'weak' users of the network.

- ***Economical***

- sustain the economical development and the elimination of the diseconomies;
- increase the competitiveness of the urban areas;
- arising of new business ventures;
- reducing the energy consumption;
- sustain the e-commerce reducing the delivery time;
- address the social costs of the transportation on the interested subjects.

- ***Environmental***

- reduce air and noise pollution;
- protect highly populated areas.

- ***Urban***

- requalify historical centers;
- preserve the presence of the commerce and craftsmanship in the city center;

- optimise the usage of vehicles and infrastructure and consequently reduce delivery times and length of trips;
- protect buildings, especially in the historical centers;
- reduce number and lengths of private movements;
- reduce the concentration of the deliveries in several time slots;
- sustain the commercial concentration.

1.3 City Logistics measures and strategies

The main strategies adopted in City Logistics for the achievement of the above presented targets can be classified as follows:

- rationalization of the freight flows in the urban areas and realization/strengthen of the infrastructures;
- usage and installation of telematic technologies and ITS systems for the Infomobility;
- access limitation measures;
- road pricing measures;
- usage of City Distribution Centers.

The first two strategies will be explained in detail in the following. Other fundamental concepts, issues, trends, and challenges of City Logistics may be found in Russo and Comi [87], Taniguchi et al. [93], and the proceeding books of the City Logistics conferences available through the Institute of City Logistics web-site [57], as well as the websites of the projects Trendsetter [98], CITY PORTS [30], BESTUFS [14], the CIVITAS Initiative [31], etc.. Here a brief discussion about the other ones is provided [73]:

- Road-pricing: it consists in the payment of a tariff for the vehicles that move within the city. Basically road pricing is a tax on the congestion and it is computed taking into account the external costs and its negative externalities. It can improve the status of the network in terms of congestion, but it does not affect significantly the emissions. In fact this tax motivates the usage of smaller vehicles, but if they are not applied together with other measures related to the rationalization of the distribution activities, it would provide just an increase in the amount of travels with smaller vehicles. Moreover it is difficult to define its value. In fact a too much lower tax would have insignificant effects and if too high they could have the effect of the relocation of several activities. Road pricing, at the moment, had a scarce application, basically for the difficulty in defining fair tariffs, based on time windows, urban zones, vehicle characteristics, loading factor for the commercial vehicles, user of the network (residents and freighters).

- Park pricing: It consists in the payment of a tariff for the parking based on the time windows, parking time, areas and reasons of the parking and user categories. It arises with the aim of charging the public space, which is considered as a limited resource. This measure is highly diffused for the regulation of the private mobility, but not much has been done for what concerns the freight vehicles.

- Urban freight flows regulation: the main measures used for the control and regulation of the traffic flows can be classified as follows:

  - definition of speed limit, rights of way, one way streets;
  - definition of limited traffic zones and pedestrian areas;
  - forbid the circulation of pollutant vehicles and motivate the usage of environmental friendly vehicles;
  - penalize the private transportation;
  - definition of time windows for the access in the urban areas and for loading/unloading operations;
  - usage of preferential lanes or of predefined paths for the delivery and the pick-up of goods;
  - limitation on the deliveries in several time slots;
  - limitation on the weight and size of the vehicles which perform the deliveries;
  - authorization for the entrance in the city center just to the best practice operators;

  - provide incentives for the renewal of the fleet vehicles.

- City Distribution Centers: for sure the more tangible aspect of City Logistics can be identified in the platforms for the consolidation of the flows entering and leaving the urban areas, referred as City Distribution Center (CDC) or Urban Freight Consolidation Center. Their function is the rationalization of the movements in the urban areas, consolidating in a single point the freights for and from the city. They are basically devoted to reduce the fragmentation of all the movements that do not pass through other platforms or warehousing point. Their main targets are the increasing of the loading factor of the vehicles and the improvement of the coordination among the different subjects. CDC are basically classified in function of:

  - number and kinds of offered services;
  - number and kinds of served factories;
  - number and kinds of available vehicles;
  - location, sizes, and served users;
  - kinds of subjects involved in the realization and in the management of the infrastructure;
  - integration with other public services and urban logistic .

  The discussion about these facilities will be resumed in the following. But, in first instance, it is important to underline several results obtained in national and international experiences which contemplated the usage of a CDC:

  - reduction of the number of trips from the 30% to 80%
  - reduction of the length of the trips from 30% to 45%
  - improvement of the loading factor from 15% to 100%
  - reduction of the polluting emissions from 25% to 60%

Concluding, traditionally the term City Logistics is used to indicate the set of problems and measures related to the freight management in a urban and metropolitan area, with particular reference to the location and dimensioning of the interchange centers, choice of the more opportune carriers in terms of freight typology and dimension, and the determination of the paths directed to the city center.

This definition appears limited and reductive if referred to the extreme amount of the above described “logistic” problems in a urban and regional area. Then, the term City Logistic could be extended in various ways, both regarding the spatial reference (urban, regional) and/or the “contents” of the logistics (freight transportation, people mobility etc.).

In fact the freight distribution is just one of the component which affects the social and economical life of a urban areas, but many other traffic components explicate their effect on the urban areas, such as: urban traffic, public transportation, infomobility, ambulances, tourist transportation, hazardous materials transportation, etc.. These components use the same network and therefore they cannot be considered as stand alone systems.

Chapter 2

Freight Distribution and Infomobility

In this chapter a general discussion about the two issues of City Logistics, which will be developed in the following, is provided, i.e. freight distribution and Infomobility. Each issue will be described with particular reference to its components, targets, critical aspects and relation with City Logistics strategies and decongestion of the urban areas. Moreover a brief presentation of the two optimization problems, which will be approached in the following chapters, is provided: multi-level location-routing problem for a freight distribution system and a path covering problem for the interception of the flow traversing a transportation network.

2.1 Freight distribution problem

In the last 30 years a great interest has been addressed to the freight distribution systems and related logistic problems. Nowadays freight distribution is a vital activity for all companies, urban areas and countries, since it is at the base of almost all the economic transactions which foresee the transportation of goods/products. In fact it creates a link among all the members of a supply chain located in extra-urban areas and urban areas and the final customers, represented by residents, retailers, shops, etc.. Moreover it is also one of the major source of employment. A not comprehensive classification of the urban freight flows can be the following [100]:

- freights for the industry;
- construction materials;
- goods functional to the commerce (shops, supermarkets, commercial centers);
- goods and materials used by companies which produces services;
- hazardous materials (fuels and other industrial dangerous materials);
- shipment and material movement due to the delivery companies;
- city solid garbage.

On the other side, the negative effects, deriving by presence of trucks moving on road networks, cannot be neglected. In fact freight distribution competes with private and public vehicles transporting people for the capacity of the streets and arteries of the city, and contributes significantly to congestion and other relevant externalities such as congestion, air pollution, environmental nuisances, noise, safety and intrusions. Just to give an idea of the dimension of the problem, from a recent estimation coming from the European conference of the Transportation Ministers (CEMT 2003), it arises that freight transportation represents the 30% of the total transferred tons for travelled chilometers (*txkm*), and moreover it represents the 20% of the equivalent vehicle traffic and the 60% of the pollution coming from complex powders. In a recent study, it arises that the 25%-30% of the freight transportation in the European cities uses the 20-35% of the available street capacity. In Italy the 10% of the energy consumption is used for the freight transportaion and moreover it is the cause of the 10% of the pollution of the overall produced pollution.

Therefore, the main causes of the high inefficiencies and delays of the transportation activities can be individuated in:

- congestions levels of the urban areas where vehicles devoted to distribution have to move;
- lack of dedicated infrastructures and parking areas;
- low level of loading factor of vehicles;
- just-in-time policies and e-commerce

Concerning the last point, it is important to underline that the diffusion of just-in-time strategies and e-commerce is the cause of the fact that many deliveries of small dimension (also to the same destination) have to be performed to have more punctual deliveries. This means having a great number of almost empty vehicles traveling in the urban areas.

City Logistics is aimed at the planning, organizing, controlling and coordinating the urban freight flows and the related information flows, or more generally its aim is the rationalization and the optimization of all the activities that take place within the urban limits, in order to improve the liveability and the accessibility, without contrasting and/or delaying the social, environmental, economic and financial development of the urban areas. Hence its main targets, with reference to freight distribution, can be summarized as follows:

- reduction of air pollution and emissions which influence climate change;
- reduction of traffic noise;
- improvement of general safety;
- reduction of other forms of nuisance such as risk, physical hindrance and vibration;
- reduction of the consumption of urban space for transport infrastructures and delivery points;
- slowing down the exhaustion of natural resources, such as materials and fossil energy.

The guiding lines of a City Logistic policy for freight distribution are based on the following main points:

1. better fleet management practices, that means increasing the average loading factors of trucks and consequently minimizing the empty trips;
2. rationalization of distribution activities and traffic regulation(road pricing, definition of pedestrian areas, limitation on the size of trucks entering in the urban areas etc.);

3. freight consolidation different shippers and carriers using the same environmental friendly vehicles;

4. co-ordination of operations at all city levels;

5. deployment of intermodal infrastructure and definition of corridors for the freight transportation;

6. usage of environmental friendly vehicles.

All these strategies have to be adopted in an integrated way, because the usage of just a part of them would vanish the effects of the others. This is particularly clear for what concerns the definitions of regulations and restrictions. For example imposing limitation to the size of trucks entering in the urban areas, if not integrated with a rationalization and consolidation policy of the freight flows, would just cause an increase of traffic due to small trucks; similarly restrictions on the number of trucks in the urban areas would have negative effects on the economy if not integrated with fleet management policies and the usage of ad hoc infrastructure; and finally, strict regulation could cause the relocation of industrial and commercial activities in less constrained areas.

Moreover better fleet management practices could partially address this problem. But only partially, since it would concern individual carriers or shipper-customer combinations only. As indicated in most of the City Logistics literature, significant gains can only be achieved through a streamlining of distribution activities resulting in less freight vehicles traveling within the city. The *consolidation* of loads of different shippers and carriers within the same vehicles associated to some form of *coordination* of operations within the city are among the most important means to achieve this rationalization of distribution activities. The utilization of so-called green vehicles and the integration of public-transport infrastructures (i.e., light rail or water canals) may enhance these systems and further reduce truck movements and related emissions in the city. But consolidation and coordination are the fundamental concepts of City Logistics.

Obviously the cooperation of all the actors of the freight distribution system (shippers, freight carriers, final customers and local government) is a key element for the success of such measures.

Therefore it is necessary to efficiently solve the paradox arising in this situation, i.e. society is not well accepting truck within urban areas, but at the same time, it represents the greatest source of demand for the distribution.

To this aim it is necessary to design a freight distribution system for urban areas which has to be efficient not only from the economic point of view, but also from the environmental and social point of view. The goal is to reduce the impact of freight transportation on the city living conditions, reduce congestion and pollution, increase mobility, improve living conditions, and, in general, contribute to reach the Kyoto targets for emission reductions (the spirit of the accord, at least), while not penalizing the city center activities. More precisely, one aims to reduce and control the number and dimensions of freight vehicles operating within the city limits, improve the efficiency of freight movements, and reduce the number of empty vehicle-km.

2.1.1 Decisional levels and stakeholders

The problem of designing and/or optimizing a freight distribution systems concerns three different decisional levels and involves different stakeholders [93]. Concerning the first point, three decisional levels are generally considered:

- Strategical level: it concerns decision which foresee relevant investments, therefore long term decisions, whose planning horizon generally is of several years. Basically it involves decision concerning the type, the location and the number of facilities to open, the choice of transportation modes to adopt and their evaluation from an economic and financial point of view.

- Tactical level: it concerns decision on medium-term time horizon. Basically in this phase it is addressed the problem about how to use the resource that we have in our distribution system in a generic, or most probable, scenario. Therefore in this phase decision about scheduling of resources, routing of vehicles, etc. are defined.

- Operational level: it concerns the operations that have to be performed over small time periods, or in other words real-time operation. The length of the period depends on the specific problem under investigation, it could be minutes, or days. Basically it is aimed to planning the distribution at the lowest level, that means for example, the allocation of the human resources or re-routing of some vehicles. Decisions have to be taken in a dynamic context and their evaluation has to be performed hypothesizing different scenarios.

Concerning the second point, three key stakeholders in freight transportation can be individuated [93]:

- Shippers: they can be considered as the customers of the freight carriers. They want their products to be delivered on time to the final customers, generally represented by other companies, retailers/shops and people. Generally their aim is to keep their level of service as high as possible and to satisfy the demand at the minimum cost. Moreover, often, their inventory policies are based on just-in-time paradigm, that means it is aimed to have low level of stocks and therefore demand satisfaction within specified time windows implies the usage of just-in-time transportation systems. Obviously their decisions on using the services of a certain freight carriers or another one, is based on the price and other factors like temporal constraints or reliability.

- Freight carriers: they actually perform the distribution with the aim of maximizing their own profit. Therefore they organize the distribution process providing and managing unimodal or multimodal transportation services, moving among the infrastructure facilities (hub, air terminal, rail terminal, intermodal platforms, etc.) until the final customer. Generally they use different transportation modes for long-haul transportation and different smaller vehicles for the distribution in smaller areas (urban areas). In the second case they have to face directly the problem of using road network, characterized by many operative constraints and high level of congestion, which make harder the respect of time windows.

- Final customers: they are the people, retailers, shops or also companies that are within a specific area and that represent the demand for the shippers. They want products/goods to be delivered on time, but they want also that the impact of the freight distribution in their areas to be minimized.

In this context governments and infrastructure providers often coincide, even if in the last years the management of the most important network infrastructure have been committed to private companies, under central government control (this process is quietly diffused in Europe). They provide the distribution infrastructure (facilities, road network, railroad network, etc.) and are in charge of the regulations and of the economic policies on territories under their control. Moreover national and local governments have a very hard target, in fact, both, for long-haul transportation and for distribution in smaller areas (like urban areas), they are aimed at minimizing the effects of the freight distribution, without contrasting the economic development. These two targets involve different actors and its easy to understand that they are in contrast and therefore it is necessary to solve the hard related trade-off. Often they act through the introduction of regulations for all stakeholders involved in the system.

Since the above discussion about the different planning decision and stakeholder categories, the design of an efficient freight distribution system and/or the optimization of its performances for large and small areas is a very challenging problem, where it is important to keep into account all the variables and contrasting targets of the different stakeholders.

2.1.2 Single-echelon freight distribution system

Most contemplated and initiated projects are implementing some form of *single-echelon* system where transportation to and from the city is performed through facilities called *City Distribution Centers* (*CDC*; the terms *Intermodal Platforms* and *Logistics Platforms* are also used) located at the city limits.

Single-echelon systems do not appear interesting for large urban zones, however. More general two-echelon systems, combining major CDCs and satellite platforms strategically located within the urban area, appear promising for such cases (Crainic et al. [34], Gragnani et al. [49], Crainic et al. [35]). A city distribution center is thus a facility where shipments are consolidated prior to distribution. It is noteworthy that the CDC concept as physical facility is close to that of intermodal logistic platforms (and freight villages) that link the city to the region, country, and the world. Intermodal platforms receive large trucks and smaller vehicles dedicated to local distribution, and offer storage, sorting, and consolidation (de-consolidation) facilities, as well as a number of related services such as accounting, legal counsel, brokerage, and so on. Intermodal platforms may be stand-alone facilities situated close to the access or ring highways, or they may be part of air, rail or navigation terminals. The city distribution center may then be viewed as an intermodal platform with enhanced functionality to provide coordinated and efficient freight movements within the urban zone.

CDCs are thus an important step toward a better City Logistics organization and they are instrumental in most proposals and projects so far(e.g. Browne et al. [18], van Duin [39], Taniguchi et al. [93], Thompson and Taniguchi [94]). Most City Logistics projects were undertaken in Europe and Japan and involved only one CDC facility and a limited number of shippers and carriers.

2.1.3 Two-echelon freight distribution system

The CDCs certainly have improved the freight distribution in urban areas in the last years, but the initial success of the related system has showed some deficiencies for what concerns its usage in big cities, where the freight flows have increased significantly in the last years and the trend is not going to change. The reasons at the base of this situation are:

- CDCs located rather far from the center. If the aim is to minimize the number of trucks in the urban areas, then heavy truck should be used in order to consolidate on the same vehicle as many orders as possible. This implies that there will be large trucks moving within the urban areas, performing long routes to serve all the final customers, with difficulty in respecting the delivery time-windows.

- The particular structure of city center of big cities. Big cities are very constrained areas not only for what concerns the density of population and the variety of land utilizations, but especially for the road network infrastructure, characterized by a wide variety of streets of different width, one way streets, few and limited zones for parking, interdicted zones to the trucks etc.. For these reasons in the last years new structures for freight urban distribution system have been contemplated, based on the utilization of more than one intermediate facilities. The idea that we contemplate is based on the expansion of the concept of CDC.

Two-echelon systems have been recently proposed for such cities (Crainic et al. [34], Crainic et al. [35], Gragnani et al. [49]).

The two-echelon City Logistics concept builds on and expands the CDC idea. City Distribution Centers form the first level of the system and are located on the outskirts of the urban zone. The second echelon of the system is constituted of satellite platforms, *satellites* for short, where the freight coming from the CDCs and, eventually, other external points may be transferred to and consolidated into vehicles adapted for utilization in dense city zones. Satellites perform limited or no vehicle-waiting or sorting activities, vehicle synchronization and transdock transshipment being the operational model. This point is fundamental for this idea of distribution system, since in this way at satellites no special infrastructures and functions have to installed, but existing facilities can be used, like for example underground parking slots or municipal bus depots, or spaces like city squares and therefore no high additional costs have to be sustained (Crainic et al. [34]) for satellite activities.

Two types of vehicles are involved in a two-tier City Logistics system, urban-trucks and city-freighters, and both are supposed to be environmentally friendly. *Urban-trucks* move freight to satellites, possibly by using corridors (sets of streets) specially selected to facilitate access to satellites and reduce the impact on traffic and the environment. Moreover, since the goal is to minimize the truck movements within the city, rules may be imposed to have them travel as much as possible around the city, on the "ring highway"s surrounding the city, and enter the city center as close to destination as possible. Urban-trucks may visit more than one satellite during a trip.Their routes and departures have to be optimized and coordinated with satellite and city-freighter access and availability. *City-freighters* are vehicles of relatively small capacity that can travel along any street in the city-center area to perform the required distribution activities. City-freighters may be of several types in terms of functionality (e.g. refrigerated or not), box design, loading/unloading technology, capacity, and so on. Efficient operations require a certain standardization, however, so the number of different city-freighter types within a given City Logistics system is thus assumed to be small. This should be determined during the system design and evaluation phase.

From a physical point of view, the system operates according to the following sequence: freight arrives at an external zone where it is consolidated into urban-trucks, unless it is already into a fully-loaded urban-truck; each urban-truck receives a departure time and route and travels to one or several satellites; at a satellite, freight is transferred to city-freighters; each city-freighter performs a route to serve the designated customers, and then travels to a satellite (or a depot) for its next cycle of operations. From an information and decision point of view, it all starts with the demand for loads to be distributed within the urban zone. The corresponding freight will be consolidated at external zones yielding the actual demand for the urban-truck transportation and the satellite transdock transfer activities. These, in turn, generate the input to the city-freighter circulation which provides the last leg of the distribution chain as well as the timely availability of empty city-freighters at satellites.

Obviously in this system we will have an increase of the costs for the additional transshipment operations which were not performed in a single-echelon system. Anyway these costs will be compensated, even if just in part, by the consolidation of the freights and the decrease of empty trips and by the economy of scale that will arise for the distribution activities. In figure 2.1 a representations of the two-echelon freight distribution system is shown.

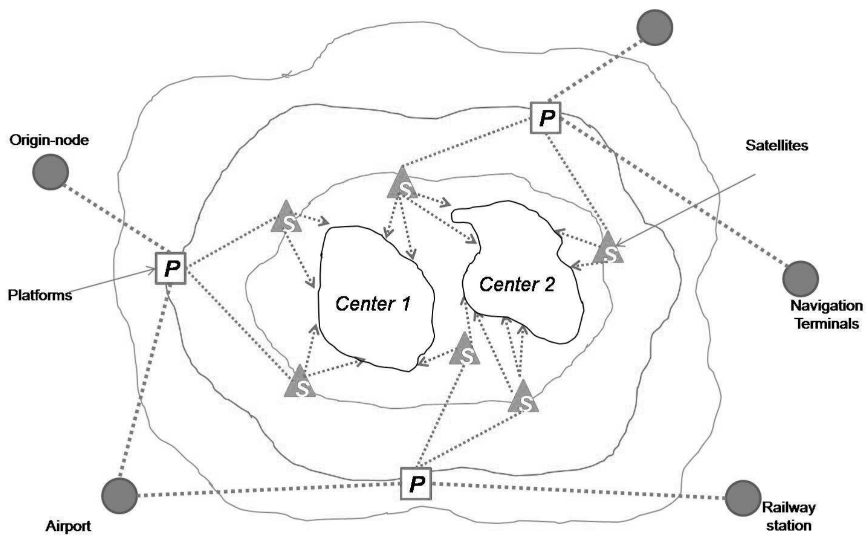


Figure 2.1: Two-echelon distribution system.

2.1.4 Optimization for freight distribution

The planning of a freight distribution system is a very hard problem. In fact it involves different stakeholders and decisional levels. In fact strategical decisions, as for example the location of the facilities and tactical decisions, as for example the definition of the routing of the vehicles, have to be taken.

In Operation Research literature the three different planning levels are generally treated separately, and just in few cases integrated approaches have been considered. Main problems for each decisional level have been assimilated to well known problems, widely treated in literature from the modeling and algorithmic point of view. Strategical decisions often involve problems of facility location (FLP) and service network design (SND), tactical decisions involve vehicle routing (VRP) and scheduling problems, and finally operational level decisions involve particular variants of the VRP, scheduling and assignment problems.

In the following we will concentrate on the integrated strategical-tactical design problem of a two-echelon freight distribution system. The aim is to define the structure of the above presented two-echelon systems for freight transportation, aimed at optimizing the location and the number of the two different kinds of facilities (platforms and satellites), the size of the two different vehicle fleets (urban trucks and city freighters) and the related routing on each echelon.

Therefore the problem has been modeled as a two-echelon (multi-level) location-routing problem (2E-LRP). These problems are NP-hard since they arise from the combination of two problems which are NP-hard as well, i.e. FLP and VRP. At the best of our knowledge, multi-level location-routing problems have not yet been addressed either with exact or heuristic methods.

2.2 Infomobility

Urban and regional travel demand has reached a dimension not compatible with the space capacity of urban centers and with the environmental protection needs. High traffic flows induce relevant pollution phenomenon, accidents and diseconomies in good and service production field. To decrease traffic congestion levels in urban areas it is necessary to adopt adequate policies, on one hand, for travel demand management (TDM) and control to reduce the use of private transportation, and on the other hand for the optimal management of available transportation supply. In this context the *Intelligent Transportation Systems (ITS)* are particularly relevant [104]. They apply advanced technologies, proper of computer science and telecommunication, to traffic and transportation management. In this way they give information about traffic condition, in terms of relevant events or congestion level, to the road network users. In the information society of the third millennium the transformation process towards a sustainable mobility depends on the ability to collect, elaborate and distribute in the best way information about traffic network status and available transportation services. For this reason *Infomobility* assumes a relevant importance. The definition of Infomobility (*ICT, Information and Communication Technologies*) can be summarized in three main points:

1. Real time information to the users about the congestion of the network system.

2. Communication of the information anytime, anyplace and anywhere.

3. Usage of technologies for the intelligent management of the mobility system.

The main reasons at the base of Infomobility arising can be individuated in the great increase of traffic in the road network system and the consequent need of having updated and real time information on the congestion phenomenon in order to reduce its negative effects. The main application fields of these technologies can be summarized as follows:

1. planning the routing of the vehicles and their scheduling;

2. localization of the vehicles;

3. automation of the guide and of the handling;

4. tracking of the freights;

5. exchange of the information among the different urban logistic actors;

6. management of the logistic flows deriving from the on-line market;

7. traffic light and electronic regulation;

8. support to the economic measures, as road and park pricing;

9. optmization of the parking for loading/unloading operations.

Therefore ICT involves the application of advanced technologies to help reduce the costs of transportation systems and makes "skinful use of advanced electronic and communication technologies to merge people, vehicles and roads into integrated, intelligent systems".

ICT in the last years have been widely used in support of City Logistics systems, and therefore they constitute a new dimension for the problem. The main impacts of the ICT on City Logistics can be classified in four main categories [93] :

- E-commerce:
  - change in the supply chain structure;
  - increase of the length of the trips;
  - increase of the number of deliveries;
  - time sensitive services for the clients;
  - increase the number of carriers performing the distribution;
  - exploitation of delivery and pick up points in the network.
- E-logistics:
  - increase in the commercial competitiveness levels;
  - development of cooperative delivery systems based on internet and intelligent transportation systems (*ITS*).
- E-fleet management:
  - traffic monitoring through the use of GPS technologies;
  - tracking of containers and pallets;
  - planning of the trips in order to avoid congestioned areas through the usage of digital map and real-time traffic information;
  - definition road-pricing strategies;
  - improve the effectiveness of the transportation system.

ITC are based on two key elements, intelligence and integration. Intelligence involves gaining knowledge through data collection and information processing. Integration relates to connecting and co-ordinating the key components of the system.

Therefore the term Infomobility indicates the set of technologies and procedures which provide the required information to managers and customers in order to obtain an efficient mobility of private and public transport. Development of communication networks and their applications in the transport and traffic field create new opportunities for the management of the transport system at urban scale, which can be identified in:

1. realization of monitoring systems and automatic data collection systems;
2. realization of navigation system on board;
3. mobility management;
4. realization of pre-trip informative systems;
5. realization of en-route informative systems;
6. realization of on-board systems for the assistance and safety during the guide.

Infomobility services are finalized to environmental, safety and efficiency purposes. In the following the main effects of ITC systems for these issues and several results are reported [1], [17].

For the environmental aspects it has been calculated that variable message information and radio services determine transport costs reduction and consequently the reduction of pollutant emissions (carbon monoxide, hydrocarbon). Infomobility in public transport management improves its supply and contributes to the air pollution reduction directly, with the reduction of travel times, and indirectly obtaining an higher demand for public transport. Telematic system which control accesses to the limited traffic areas can reduce pollution in those areas until the 50%. From a study on several Infomobility projects the following results arise:

1. variable massage signs systems determined a reduction of the 20% in the delays due to the traffic and a reduction of the 10% of carbon monoxide, 5% of hydrocarbon and 5% of nitrogen oxide

2. the route guidance system allow to save the 5% of the energy consumption

3. the systems for the management of public transportation determined an improvement of the offered services and determined a decrease of the related polluting emission of the 4.4% for year.

4. telematic systems for the control of the access provided a reduction of the 50% of the pollution in the zone under control

In terms of efficiency, control strategies (based on the information provided by the monitoring system about the traffic flow at the entrance or the exit of extra-urban road) determine increases of the average speed on the highways. The automatic systems for the payment of road toll allow to considerably reduce the travel times on highways. The traffic management through the automatic adaptation of traffic lights times and the consequent reduction of the waiting times, the information to the travelers about the available public transportation options, the congestions of the main roads, the parking availability and the priority for the public transport by means of an intelligent traffic lights system, determine a meaningful reduction of the travel times for all kinds of transportation. The management and control systems for the freight transportation vehicles, based on data transmission services, produce a reduction of travel times, of delivery service times and of the covered distance, due to a dynamic routing optmization procedure for the definition of paths. From a study on several Infomobility projects the following results arise:

1. the variable message signs provided a decrease of the delays of about the 20%

2. the Ramp metering, systems used for the monitoring of the traffic on the main roads determined an increase of the average speed of 21% on the highways and of the 16% on the arterial streets and of the 19% on the ramps

3. the automatic payment systems determined a decrease of 40 hours for year for the users who travel on the highways for at least 4 days a week

4. management of the urban mobility systems operating on the light traffic control system, on the vehicular flow, on the information to travelers, on the parking slots availability etc.

5. the Radio data system Traffic message channel allows a decrease of the 3.9% on the traveling time

6. the Route guidance system allows to save the 4.8% of the traveling time

7. the system for the management and control of the fleets based on telematics, allows a save of the 5% on the traveling time, 12% on the delivery time and 6% on the travelled distance.

Finally for what concerns safety, benefit are achievable from variable messages through additive information concerning weather and traffic conditions. A reduction of maximum speeds and of accidents in fog and rain conditions can be observed. Monitoring systems installed on commercial vehicles also determine a reduction of the number and risk of accidents. Emergency calls based on satellite technologies and mobile phones enable the reduction of arrival time of emergencies vehicles in case of accidents. From a study on several Infomobility projects the following results arise:

1. the variable massage signs systems, integrated with a system for the management of the information to travelers en-route, reduced the level of accidents of the 30% and the number of murders and injureds of tge 40%, the maximum speed of the 10%, the accident in rain condition of the 30% and the accident in fog condition of the 85%

2. the systems for the emergency calling based on satellites usage and cell phones determined a reduction of the response time of the 43% and consequently an increase of the survival rate of the 7.12%

To conclude we can say that Infomobility instruments had a wide spread in extra-urban area, but not in the city, hence they do not entirely employ their potentiality in terms of traffic decongestion. The content of the information plays a relevant role in infomobility services and could play a role still more important in the reduction of the levels of congestion of a urban area.

2.2.1 Infomobility and information

At this point it is clear that the key element for the Infomobility services is represented by the *information*. The information can be defined as the massage transferred from a subject to another using a physical support. In fact Infomobility means the activities aimed at informing the users of the network and of transportation system about the status and the accessibility of the network.

The information can be classified considering different criteria [103]. Concerning the state of the network we can have:

- descriptive information: in this case the information provide information about the state of the network, focusing on the traveling time on a predefined path, on the congestion of several links or on the occurrence of special events, etc..

- predictive information: in this case the information provide advices about the way to perform a trip, i.e. they give information about the path or the mean of transportation to choose, etc.. Obviously these advices can or cannot be accepted by the users, depending on the confidence level addressed to the information and on the reliability of the informative system.

The information can also be classified depending on the way it can be achieved by the users:

- pre-trip information: if they are available before the beginning of a trip and they concern all the possible choices in terms of paths, time and mean of transportation

- en-route information: if they are available during the trip.

The information can be classified considering a time scale.

- historical information, if they are referred to the previous observation on the state of the network

- current information, if they are referred to the actual state of the network

- predictive information, if they provide information about the future state of the network starting from a predictive analysis of the data collected in the time.

Finally, information can be classified in terms of usability by the users:

- passive information, if they are involuntarily achieved by the users
- active information, if the users voluntarily look for the information

The information just classified are then processed in order to be profitably used. It is not easy to have a clear classification and a schedule of the operations performed by the infomobility on the achieved information, because they are highly integrated. Anyway the following steps can be considered:

- data collection on the state of the network (performed by traffic counters, sensors, camcorders, intelligent gate, electronic payment counter, webcam, etc.);
- management of the information;
- elaboration of the information;
- diffusion of the information to all the users.

2.2.2 Infomobility systems

In this section a brief discussion about most known Infomobility technologies is provided.

Info points: these systems can be considered as discretionary services. In fact it is the user that has to decide when and where to achieve information by this technology. Info-points are stand alone systems that the users can find along their trip. The information provided by these systems are related to alternative paths, presence of fuel station and parking areas. These systems use basically two kinds of technologies: touch screen and vocal technology. Moreover they can ba classified depending if they are wall installed or onshore.

Sensors: traffic sensors are aimed at measuring the intensity and the vehicle spatial distribution, providing in this way the information required by the logic system to choose the opportune strategy. We can have to kinds of logic system.

- on line: in this case the logic system on the basis of the achieved information choose the best strategy to perform. Therefore in this case the human operator has only a supervision function

- off line: in this case the logic system on the basis of the achieved information send them to an operator and the possible strategies to put in act.

Inspection stations: with this expression we refer to all the instrument devoted to take under control several important impacts, as for example the control of the air pollution and its quality. They are mainly located in two kind of areas, urban and suburban. In the first case, they are generally located in vicinity of highly congested areas. For what concerns instead the second case they are generally located in vicinity of industrial areas or of main roads.

Variable Message Signs: they represent the most popular infomobility instrument. They are devoted to traffic control and are used to provide the information and indication to the drivers during a trip. The use of these systems has effects on both the safety and for the information to the users of the road network when events that modify the normal state of the network occur. The information is displayed in real time and it can be controlled or by a remote central or by a local station. The information message must be short and concise in order to be easily and rapidly achieved by the travelers and consequently take a fast choice, without being distracted during the reading. Therefore we can say that the VMS are thought just to act on the behavior if the travelers in a way that it improves the flows and traffic operations. They can manage the access at tunnels, bridges and highways intersection and they help the drivers to take the right decisions during multi-choice paths, or in case of roadworks, where the conditions of the network can rapidly change. The VMSs signs are used to communicate several kinds of information, related to alternative paths for critical point of the network during several time windows, interrupted lanes, roadworks, congestions, accidents, availability of parking slots, suggested paths to reach local attraction and sport events, suggested speed, traveling time on several streets. Therefore we can say that VMSs are very important instruments for the collective route guidance. In fact the provided information allow to re-establish the flow equilibrium of the network. There are different kinds of variable message signs, which can be classified with several criteria, as for example the technological characteristics. Anyway the main difference among the different kinds of VMS is in the fact that we have fixed VMS or mobile VMS. The last ones are based on the idea that they could be transferred from a location to another, operating for short periods and then moved in the area where an emergence occurs. The fixed VMSs allow to display information and suggestion which can be easily seen and read from high distances and which can be updated in real-time. These instruments provide information related to the traffic condition and viability, traffic deviations, parking slot availability, pollution of the air and potential closure of the street to the traffic, special events, public utility messages, emergences and indication related to the usage of public transportation. They are located on the highways, ring roads, main roads for the access to the city, highly used urban streets, vicinity of crossroads for the traffic deviations, squares, traffic controlled zones, historic centers, pedestrian zones and crucial points. They have to be located in the most strategic points in order to provide information to the highest number of users of the network. The effectiveness of the VMS depend on the time that the users have to read, the speed of the users and distance between the location of the instrument and the point where it is noticed. The mobile VMSs are generally installed on vehicles and they represent an important informative system in case of accidents, road maintenance, fog, etc.. In fact it is easy to move the VMS in the interested area and to place it, for example, one or several kilometers before of road accidents, fog banks, open yards and in general indicate problems to the viability.

2.2.3 Optimization for infomobility

Urban and regional travel demand has reached a dimension not compatible with the space capacity of urban centers and with the environmental protection needs. High traffic flows induce relevant pollution phenomenon, accidents and diseconomies in good and service production field. The increase rate of travel demand in some regions is so high to make ineffective or just sufficient the structural operations aimed to increase mass or private transportation supply. In this context Infomobility services are particularly relevant. They apply advanced technologies, proper of computer science and telecommunication, to traffic and transportation management. In this way they give information about traffic condition, in terms of relevant events or congestion level, to the road network users. In this context the thesis studies the problem of optimal location of infomobility services, with particular reference to the Variable Message System (VMS) for the route guidance of vehicular flows. These facilities are in general referred as *flow intercepting* *facilities*. These facilities can be used by the flow units of the network or proposed to/imposed on them along their pre-planned path from an origin to a destination. In other words, the purpose of the movement is not to obtain a service, but if there is a facility on the pre-planned path, the flow units may choose to interrupt the journey to obtain the service, before continuing their path.

The available models are quite general and so it is necessary to formulate specific constraints for the described problems. In particular it is necessary to insert the model in the context of the estimation process of the origin/destination travel demand matrix. It is clear, in fact, that flow monitoring systems have to be located in order to maximize the likelihood of the estimated O/D matrix, with respect to the real O/D matrix.

Part II

Two-echelon location-routing problem 2E-LRP

Chapter 3

Location-routing problem definition

In this chapter the generalized location-routing problem (LRP) is presented and the inappropriateness of approaching this problem with classical facility location models and methods is discussed. Then a more precise definition and a formulation for the generalized LRP are given and an expression which allows to synthetically classify LRP on more than two levels is provided. Finally the basic assumptions of LRPs in the context of freight distribution system design are presented.

3.1 FLP and LRP

Single and multi-level facility location (FLP) models have been widely adopted in literature to represent freight distribution systems. Generally, but not in all cases, in these models the transportation costs are assumed to be a linear function of the *straight-line* (also referred as *radial distance*) between a facility and final customers. This is based on the assumption that each customer is served by a full load truck, which performs a dedicated route, whose transportation cost is well approximated by the straight-line distance from the facility. In this context the objective function is aimed at minimizing the sum of assignment/transportation costs and it is expressed with the following equation:

$$\sum_{o \in O} \sum_{z \in Z} c_{oz} \cdot x_{oz} \tag{3.1}$$

where:

O is the set of possible location for facilities

Z is the set of customers to be served

c_{ij} is the transportation cost from facility i to final customer j

x_{ij} is the quantity shipped from facility i to final customer j

Therefore it is assumed that each customer is served in a straight-and-back way and the possibility of operating multiple-stop routes is not taken into account. This assumption proved to be very efficient in context where the future demand of the customers are not known or are highly variable, but on the other side it has some limits. In fact there exist several practical situations where the approximation of routing costs with assignment costs significantly affects the performances of the system and therefore it is important to take into account all the decision variables and their interdependency, i.e.: locating one or more facilities affects the allocation of vehicles to the facilities and the length of the routes including more customers (multi-stop routes) at the same moment.

Hence in these cases location and routing decisions are strongly interrelated and have to be modeled and optimized simultaneously. The inappropriateness of approaching them through pure location models, using equation 3.1 to approximate the multi-stop routing costs, has been pointed out in several papers (Webb [102], Christophides and Eilon [27], Eilon et al. [40], Wren and Holliday [105], Perl and Daskin [84], Salhi and Rand [88], Chien [25]). In particular Christofides and Eilon [27] define a limit condition for the approximation of the multi-stop routing costs with the straight-line distance function.

They considered the case of n customers randomly and uniformly distributed in a square of side a, that have to be served on non-intersecting route, performed by m vehicles based at the same facility in the square. If z^* is the optimal total length of the m routes and z_R is the sum of the radial distances between the customers and the facility, it was showed that:

$$z^* \approx A \cdot m \cdot z_R/n + B\sqrt{a}\sqrt{z_R} \tag{3.2}$$

where A and B are two constants related to the facility position. By this equation we can say that when the first term dominates the second, i.e.:

$$z_R >> a\left(\frac{Bn}{Am}\right)^2 \tag{3.3}$$

Then we can approximate z^* with z_R. On the other side when this condition is not satisfied, then we need to model the problem as *Location-Routing problem* (*LRP*) to address location and routing aspects at the same time. LRP are particularly suitable in situations where the configuration of the routes to serve the customers are quite stable and known or when the location and routing costs are comparable in a certain time horizon. Therefore *LRP* try to overcome the limits of facility location problems integrating the location and routing aspects.

Finally we can say that location-routing problems integrate different decisional levels: location of facilities and allocation of customers are strategical decisions; fleet sizing and route definition to serve the customers are instead tactical decisions.

3.2 Generalized LRP

A multi-level freight distribution system is composed of several layers and the products flow from the top level to successive ones until the final customer. In the most frequent case we have three layers (Laporte [61]), identified respectively as *primary facilities* p, $p \in P$, *secondary facilities* s, $s \in S$, and *final customers* z, $z \in Z$. In the following we will refer to a sub-system composed of two successive layers of the whole system with the term *echelon* and therefore we will use likewise the expression *multi-level* or *multi-echelon* location-routing problem.

The location-routing problem (LRP), in its usual form, is related to the single-echelon location-routing case, i.e. it consists in determining:

1. the location of a certain number of secondary facilities s, eventually with limited capacity K_s, over a set of feasible location S;

2. the number of vehicles to use for the distribution;

3. the route that a vehicle, located at a specific depot, has to perform to visit a subset of z final customers over the complete set of customers Z.

Each final customer has a demand D_z and each vehicle v, $v \in V$, is characterized by a limited capacity UV. The costs of the system are given by the location costs of the secondary facilities H_s, transportation costs CT_{ij} and costs for the usage of a vehicle TCV. Obviously these costs have to be scaled down so that they can be referred to the same time horizon. Products are always available at the secondary facilities, and therefore routing decisions for the first echelon, i.e. between primary and secondary facilities, are not considered. The following set of variables have to be defined:

$x_{ij}^v = \{0,1\}$	*1*, if i precedes j in the routing of the second echelon, performed by city freighter v *0* otherwise
$w_{sz} = \{0,1\}$	*1*, if the customer z, $z \in Z$, is assigned to satellite s, $s \in S$ *0* otherwise
$y_s = \{0,1\}$	*1*, if a platform is opened at node s, $s \in S$ *0* otherwise
$t^v = \{0,1\}$	*1*, if city freighter v, $v \in V$, is used for distribution *0* otherwise

Then the generalized LRP problem can be formulated as follows:

$$\text{Minimize} \quad \sum_{s \in S} H_s \; y_s + \sum_{v \in V} TCV \; t^v + \sum_{v \in V} \sum_{i \in S \cup Z} \sum_{j \in S \cup Z} CT_{ij} \; x_{ij}^v \tag{3.4}$$

Subject to

$$\sum_{v \in V} \sum_{j \in S \cup Z} x_{zj}^v = 1 \qquad \forall z \in Z \tag{3.5}$$

$$\sum_{l \in S \cup Z} x_{lj}^v - \sum_{l \in S \cup Z} x_{jl}^v = 0 \qquad \forall j \in Z \cup S, \forall v \in V \tag{3.6}$$

$$\sum_{l \in B} \sum_{h \in \overline{B}} \sum_{v \in V} x_{lh}^v \geq 1 \qquad \forall B \subset S \cup Z, with\ S \subseteq B \tag{3.7}$$

$$\sum_{l \in S \cup Z} \sum_{j \in S} x_{lj}^v \leq 1 \qquad \forall v \in V \tag{3.8}$$

$$\sum_{h \in S \cup Z} x_{zh}^v + \sum_{h \in S \cup Z} x_{sh}^v - w_{sz} \leq 1 \qquad \forall z \in Z, \forall v \in V, \forall s \in S \tag{3.9}$$

$$\sum_{i \in Z} D_z \sum_{j \in S \cup Z} x_{zj}^v \leq UV \ t^v \qquad \forall v \in V \tag{3.10}$$

$$\sum_{z \in Z} D_z \ w_{sz} \leq K_s \ y_s \qquad \forall s \in S \tag{3.11}$$

The objective function (3.4) minimizes the overall costs, which is a linear combination of routing and location costs, given by the sum of the opening costs H_s of the facilities, costs for the use of a vehicle TCV and sum of transportation costs CT_{ij}. The constraints (explained in detail in *Section 4.2*) are related to routing decisions, capacity limits and location.

The single-echelon location-routing problem can be generalized in order to consider a distribution system with several interacting layers, all involved in the location and routing decisions. To this aim, Laporte [61] provided a definition for location-routing problem and a classification for them.

Two possible kind of routes (trips) from one layer to another are defined:

***R* routes**: return trips, i.e. trips connecting a single customer to a single facility

***T* trips**: round trips, i.e. trips connecting multiple customer and/or multiple facility.

In order to have location-routing problems two conditions have to be respected:

1. location decisions must be made for at least one layer;
2. tours must be allowed at least between two layers, otherwise if all trips are R trips, then the problem reduce to a multi-level facility location problem.

Based on these definitions, Laporte [61] introduced the following expression to represent synthetically the main characteristics of a location-routing problem: $\lambda/M_1/.../M_{\lambda-1}$, where λ is the number of layers and $M_1/.../M_{\lambda-1}$ are the kind of tours among two consecutive layers. Then for example with the expression $3/R/T$, we refer to a problem with three layers, R trips between the first and the second layer (first echelon) and T trips between the second and the third layer (second echelon).

An easy modification of Laporte expression is proposed, since we believe that further information are needed in order to better define a multi-level location-routing problem. The previous expression provides a full information about the routing decisions at each echelon, but does not provide any information about which layers are involved in the location decisions. In fact with the expression $3/R/T$ we do not know if we have to define the location of primary or secondary facilities or both of them.

Therefore previous expression could be integrated in this way: $\lambda/\vartheta(1-2-...-\lambda)/M_1/.../M_{\lambda-1}$, where ϑ indicates the number of layers involved in the location decisions and in the brackets they are specifically indicated. For example with the expression $3/2(1-2)/R/T$ we refer to a problem with three layers, two decisions location variables related to layer one and two, and R and T routes respectively for the first and the second echelon. Using this new expression, the generalized location-routing problem, i.e. the single-echelon location-routing problem, are referred as $2/1(1)/T$ problems.

3.3 Literature review

The idea of combining two decisional levels, strategical and tactical, for a transportation system dates back to the 1960 (Maranzana [74]). Anyway, in that period, the aim was to highlight just the difficulty of these problems. On the other side a greatest number of papers on generalized LRP starts to appear just from the '80s. LRP surveys have been proposed by Balakrishnan et al. [5], Laporte [61] and [62], and Min et al. [77]. The most recent one is by Nagy and Salhi [80], which provide a deeply focused discussion of problems and methods present in literature and future perspectives for LRPs. In the following a review of several papers is provided.

Or and Pierskalla [82] treat a $2/1(2)/T$ problem for the location of regional blood banking in the area of Chicago. They propose a non-linear integer programming model for the related LRP and an algorithm, based on the decomposition of the problem in four sub-problems, opportunely merged and sequentially solved.

Jacobsen and Madsen [58] and Madsen [72] treat a $3/1(2)/T/T$ location-routing problem. The aim is to optimize the newspapers deliveries in Denmark. Therefore they solve two routing problem among three layers, and they locate facilities in the intermediate level. They propose three heuristics for the problem. The Tree-Tour Heuristic (*TTH*), which exploits the property that by the deletion of an arc for each defined tour, the solution of the problem is a spanning tree with the characteristics that only first and second layer facilities have multiple successor; the *ALA-SAV* heuristic, which is a three stage procedure composed of the Alternate Location Allocation model (*ALA*) and the Savings method (*SAV*); the *SAV-DROP* heuristic, which is a three stage heuristic composed of the Saving method (*SAV*) and the Drop method (*DROP*).

Perl and Daskin [84] treat a $3/1(2)/R/T$ warehouse location-routing problem, with constraints on the capacity of the facilities, on the capacity of the vehicles and on the maximum allowable length of a route. They provide a discussion about location-routing problem and their complexity and a mixed-integer programming model. They solve the problem decomposing it in its three sub-components (subproblems) which are solved by exact or heuristic methods in a sequential way. The three problems are: the complete multi-depot vehicle-dispatch problem; the warehouse location-allocation problem; the multi-depot routing-allocation problem.

Laporte et al. [64], [66], [65], [67] were the only ones approaching some $2/1(2)/T$ location-routing problems by exact methods. In Laporte and Nobert [64] a single depot is to be selected and a fixed number of vehicles is to be used. A branch-and-bound algorithm is used. The authors note that the optimal depot location rarely coincides with the node closest to the center of gravity. Laporte et al. [66] consider locating several depots, with or without depot fixed costs and with or without an upper limit on the number of depots. For the special case of only one vehicle per depot, it was found to be more efficient to first reintroduce subtour elimination constraints (there would be no chain barring constraints) and then use Gomory cuts to achieve integrality. Otherwise, the authors recommend using Gomory cuts first and then reintroducing subtour and chain barring constraints. On the other hand, the method of Laporte et al. [65] applies a branching procedure where subtour elimination and chain barring constraints are reintroduced. Laporte et al. [67] use a graph transformation to reformulate the LRP into a traveling salesman type problem. They apply a branch-and-bound algorithm, where in the search tree, each subproblem is a constrained assignment problem and can thus be solved efficiently.

Srivastava and Benton [90] and Srivastava [89] present three heuristics for the $2/1(2)/T$ location-routing problem with capacitated vehicles. A "save drop" heuristic, where at each iteration they consider simultaneously dropping depots and assigning customers to routes developed from open depots; a "saving-add" heuristic, which is based on a similar scheme of the "save drop", but it opens the depots one by one, considering all the feasible sites closed at the beginning; a "cluster-routing" approach, which identifies the desired number of cluster and customers, and a depot is located in the site nearest to the centroid of each cluster; the routing in each cluster is achieved solving a TSP for a subset of customers, defined on their polar coordinates. They also perform a statistical analysis on the parameters affecting the solutions obtained with the three heuristics.

Chien [26] propose a heuristic procedure for the $2/1(2)/T$ uncapacitated location-capacitated routing problems, i.e. capacitated vehicles and uncapacitated facilities. The heuristic is based on two sequential steps. In the first step a feasible solution to the location/allocation problem is generated, where the routing costs are evaluated through two different estimators. Then they solve the routing problem with the generated solution of the location/allocation problem. The improvement of the routing solution is then based on the use of four operations: consolidation/change of vehicle, insertions, swappings and change of facility. Different combinations of these operations are performed using the two estimators.

Hansen et al. [52] extend the work of Perl and Daskin [84] for a $2/1(2)/T$ problem. In fact they propose a modified formulation for the warehouse location routing problem presented by Perl and Daskin, and they use the same decomposition of the problem, but improving the results of each single component, and consequently the quality of the final solution.

Bruns and Klose [19] propose a heuristic for a $3/1(2)/T/T$ LRP with limitations on the length of the routes. They used a location first-route second iterative approach, where the costs to serve the customers are updated at each iteration. The location phase is solved with a Lagrangian relaxation, whereas the routing phase with a local search heuristic.

Nagy and Salhi [79] treat the $2/1(2)/T$ location-routing problem with a nested heuristic. They propose an approach aimed at of avoiding the classical hierarchical decomposition location-first route-second. All customer sites are potential depot sites. The solution space consists of all possible combinations of customer sites. A first feasible solution is determined using a subset of the potential sites. For the location phase, the neighborhood structure is defined by the three moves add, drop and shift. Add means opening a closed depot, drop means closing an open depot and shift refers to the simultaneous opening of a closed depot and closing of an open depot. The most improving one is selected. For the routing phase customers are divided in two subsets: the nearest ones, which are directly assigned to an open depot to create the initial routes and the farthest ones, which are instead inserted in a route in function of the capacity constraints. The determined routes are the improved by a local search which include several tabu search features.

Tuzun and Burke [99] propose a two-phase tabu search heuristic for the $2/1(2)/T$ location-routing problem with no capacity constraints on the depots. The heuristic starts with the opening of just one depot. Performing location and routing moves it tries to find the best solution to serve the customers with just one depot. Then when no improvement is obtained for a given number of iterations, it adds another depot to the location solution and repeat the same operations. After a given number of add moves without improvement, the heuristic stops. The proposed approach foresee several moves for the definition of the neighborhood solutions. More precisely for the location moves, add and swap moves are considered. Whereas for the routing phase, insert and swap moves of customers are taken into account. These moves are performed in an efficient way, avoiding to explore all the possible insertions or exchanges of customers. In fact insertion moves are limited to routes assigned to nearest depots and swap moves are limited to the nearest customers.

Wu et al. [106] solve the $2/1(2)/T$ location-routing with capacity constraints for heterogeneous vehicles and depots. The problem is solved with a sequential metaheuristic approach. They first solve location-allocation problem and the general vehicle routing problem, then they are combined with a simulated annealing approach, integrated with "tabu list" concept, in order to prevent the ciclying.

Lin et al. [70] treat a $2/1(2)/T$ problem, where vehicles are allowed to take multiple trips. First, the minimum number of facilities required is determined. Then, the VRP solution is completely evaluated for all combinations of facilities. Vehicles are allocated to trips by completely evaluating all allocations. If the best routing cost found is more than the setup cost for an additional depot, the algorithm moves on to evaluating all sets of facilities that contain one more depot. The applicability of this method is limited as it relies on evaluating what may well be a large number of depot configurations.

Albareda-Sambola et al. [3] solve a $2/1(2)/T$ location-routing problem where they have a single vehicle for each depot. They define an auxiliary compact formulation of the problem, which transform the problem in finding a set of paths in the auxiliary network that fulfill additional constraints. They propose upper and lower bounds and they solve the problem through a tabu search heuristic, based on an initial rounding procedure of the LP solution.

Melechovsky et al. [75] propose for the first time a two-index formulation for the $2/1(2)/T$ location-routing problem, based on the two-index formulation for the VRP problem proposed by Fischetti et al. [42]. They propose an algorithm which starting from an initial feasible solution, searches for better solutions with a hybrid metaheuristic, which merge Variable Neighborhood Search (VNS) and Tabu Search (TS) principles. Therefore the key element of their approach is the integrated use of the two methods, which they realize replacing the local search procedure in the VNS framework with a Tabu Search algorithm.

Wang et al. [101] propose a two-phase hybrid heuristic for the $2/1(2)/T$ location-routing problem. They decompose the problem in the location/allocation phase and routing phase. In the first phase a tabu search is performed on the location variables to determine a good configuration of facilities to be used in the distribution. In the second phase ant colony algorithm is run on the routing variables in order to obtain a good routing for the given configuration. In the second phase, the routing problem is also decomposed in smaller sub-problems.

Ambrosino and Scutellá [2] study a complex distribution network design problem $4/2(2-3)/R/T/T$. They consider a problem where two different kinds of facilities have to be located in hierarchically ordered layers. The products are delivered from the first layer to the second one with R trips and different vehicles perform the distribution of products among second layer and third layer, and third layer and customers on T trips. Therefore, differently from the previous treated problems, they have to solve two-location routing problems. They propose different formulations for the problem in static ha and dynamic scenarios, extending the three-index arc formulations proposed by Perl and Daskin [84] and the three index flow formulation proposed by Hansen et al. [52]. They solve the problem on small instances with a general optimization software.

Prins et al. [85] solve the $2/1(2)/T$ capacitated location problem with a GRASP approach integrated with learning process and path relinking. They use a two index formulation for the problem, which differs from the one used by Melechovsky et al. [75] for the definition of the arc-variables. Their approach is based on two phases. A first phase executes a GRASP based on an extended and randomized version of Clarke and Wright algorithm. This phase is implemented with a learning process on the choice of depots. In a second phase, new solutions are generated by a post-optimization using path relinking.

Barreto et al. [6] propose a sequential distribution-first and location-second heuristic for the $2/1(2)/T$ problem. The method is based on customer clustering. They perform a huge experimentation with seven different proximity measures.

Chen and Ting [24] propose a three phase heuristic approach for the $2/1(2)/T$ multi-depot location-routing problem. They start solving the location/allocation problem through a Lagrangian heuristic, then they solve a VRP for each selected facility location through a simulated annealing procedure (route construction), and finally they run the simulated annealing for all the routes.

Özyurt and Aksen [83] propose a nested Lagrangian relaxation-based method for the $2/1(2)/T$ uncapacitated multi-depot location-routing problem. They consider the possibility of opening new facilities or closing existing ones. The problem is decomposed in two subproblems. The first is solved exactly by a commercial MIP solver, and the second resembles a capacitated and degree constrained minimum spanning forest problem, which is tackled with an augmented Lagrangian relaxation. The solution of the first subproblem reveals a depot location plan. As soon as a new distinct location plan is found in the course of the subgradient iterations, a tabu search algorithm is triggered to solve the multi-depot vehicle routing problem associated with that plan, and a feasible solution to the parent problem is obtained. Its objective value is checked against the current upper bound on the parent problems true optimal objective value.

From the previous literature review, we can say that the interest in location-routing problems had a great increase in the last years. These problems are very hard and therefore they are frequently tackled with heuristic approaches. Location-routing literature is at most devoted to the single echelon case, where we have locate secondary facilities and we have to take routing decisions between these facilities and final customers.

On the other side we can also adfirm that the literature about multi-level location routing problems is very scarce. The only contributions on this topic are the ones of Jacobsen and Madsen [58], Madsen [72] and Bruns and Klose [19], who treated a $3/1(2)/R/T$ problem, and the one of Ambrosino and Scutellá [2], who treated the $4/2(2-3)/R/T/T$ problem.

In the following sections we will focus on multi-echelon location-routing problem and specifically on the two-echelon location-routing problem and we will provide a discussion and several formulations for it. This problem arises from the two works of Crainic et al. [34] and Crainic et al. [35], where a two-echelon distribution systems for freight distribution in urban areas is proposed.

3.4 A two echelon location-routing problem (2E-LRP) for freight distribution

As said above, the location-routing problem is a strategical and tactical problem, where the capacitated multi-level location decisions are integrated with routing and fleet sizing decisions.

The problem that is going to be approached is the design of a two-echelon freight distribution system for a single representative product. Based on the notation previously presented, the problem can be referred as a $3/(2-3)/T/T$ problem. In the following we will indicate the facilities with the expression primary and secondary facilities or with the terms platforms and satellites, as in Crainic et al. [34], [35].

The problem is described through a multi-level network *G(N,A)*, where the node set is composed of three subsets, one for each layer: primary facilities (or platforms), secondary facilities (or satellites) and final customers. Therefore, more precisely, N is composed of the following three subsets: $P = \{p\}$ is the set of potential positions of platforms, where the first consolidation and transshipment operations are performed ($1^{st} layer$); $S = \{s\}$ is the set of potential positions of satellites, where the second transshipment operations are performed ($2^{nd} layer$); $Z = \{z\}$ is the set of final customers, whose positions and demand are fixed and known in advance ($3^{rd} layer$).

The products are available at the platforms P in limited amounts. Products are consolidated and transshipped on trucks which serve the satellites S. At satellites S product are transferred on smaller trucks and distributed to the final customers Z. We assume to exactly know the demand of the representative product for each customer and the platforms are always able of satisfying the whole demand.

The arc set represents the connections among the three different layers. More precisely the following connections from one layer to the successive ones are considered: route of type T from the primary facilities P to the satellites S, route of type T from the secondary facilities S to the customers Z. In the following these performed routes will be respectively referred as *first echelon routing*, from P to S, and *second echelon routing*, from S to Z.

The distribution at the echelons is performed through two kinds of trucks, which differ for their capacity. More precisely the transportation among the different layers is performed as follows:

urban trucks ($G = \{g\}$): they are the *first-echelon vehicles*, devoted to the distribution of consolidated demands from the platforms to the satellites.

city freighters ($V = \{v\}$): they are the *second echelon vehicles*, devoted to the final distribution from the satellites to the customers.

In 2E-LRP the size of the fleets are not given, but have to be determined so as to minimize the overall cost. The trucks belonging to the same echelon are characterized by the same capacity value, which is much higher than the maximum assignable demand. The number of trucks for each echelon is determined considering that total demand of the customers has to be between 90% and 95% of the maximum substainable load. This condition, used in Crainic et al. [36], guarantee that vehicles will be used near their maximum capacity and that feasible solutions can be found under the assumption of customer demands much smaller than vehicle capacities. To summarize the following basic assumptions are used in problem formulation:

- All the freight starts from the platforms
- The platforms and satellites are characterized by limited capacity. Obviously platform capacity is much higher than satellite capacity.
- The customers are the destinations of the freights and to each customer a demand is associated , i.e. the quantity of freight that has to be delivered to that customer.
- The demand of each customer and the demand assigned to each satellite cannot be split among different vehicles, neither at the 1st nor at the 2^{nd} level.
- The distribution of the freight cannot be managed by direct shipping from the platforms to the customers, but freight must be consolidated from the platform to a satellite and then, from the satellites, it is delivered to the assigned customers.
- An arc (i,j) is referred as 1^{st} echelon arc if both nodes are satellites or one is a platform and the other is a satellite. On the other side an arc is referred as 2^{nd} echelon arc if both nodes are customers or one is a satellite and the other is a customer.
- For both 1^{st} and 2^{nd} echelon vehicles, only one representative type of freight is considered and the volumes of freight required by different customers can be loaded in the same vehicles.
- The number of vehicles on each echelon is not known in advance. Vehicles belonging to the same echelon have the same capacity value. The capacity of first echelon vehicles is much higher than the capacity of second echelon vehicles and of satellites. The capacity of second echelon vehicles is much higher than the demand of the customers.
- 1^{st} echelon routes start from a platform, serve one or more satellites and ends to the same platforms.
- 2^{nd} echelon routes start from a satellite, serve one or more customers and ends to the same satellite.

The problem in its general form consists in the following decisions:

- *location decisions*: define number and locations of platforms and satellites;
- *allocation decisions*: assign customers to each open secondary facility and open satellites to open platforms. Obviously the allocation has to respect the capacity constraints of each open facility. The allocation for both echelons is a single source allocation, that means that satellites have to be assigned to just one platform and customers to just one satellite;
- *routing decisions*: number of vehicles to be used for the distribution on both echelons and related routes.

In figure 3.1 a schematic representation of a 2E-LRP solution is provided.

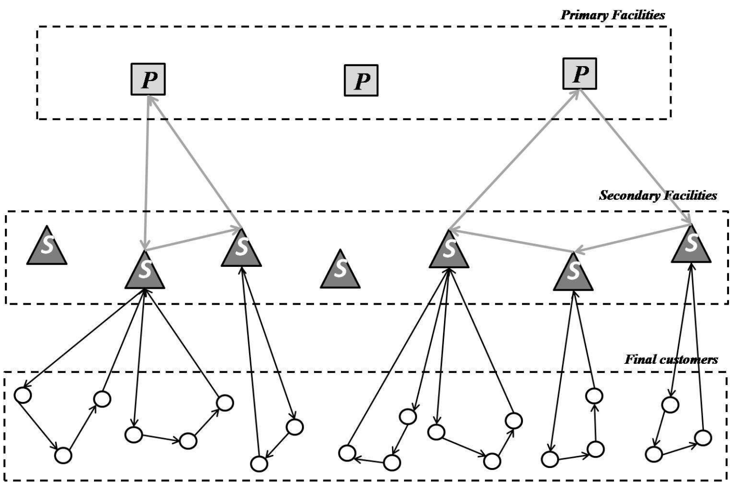


Figure 3.1: Two-echelon location routing problem representation.

The problem is easily seen to be NP-Hard via a reduction to the capacitated vehicle routing problem (CVRP), which is a special case of 2E-LRP arising when just one platform and one satellite are considered. The main issue in modeling 2E-LRP is how to connect the two levels and manage the interdependence of the different decisions between them.

Location-routing problems are clearly related to both the classical location problem and the vehicle routing problem. In fact, both of the latter problems can be viewed as special cases of LRP. If we require all customers to be directly linked to a depot, LRP becomes a standard location problem. If, on the other hand, we fix the depot locations, LRP reduces to VRP.

Chapter 4

Models for the 2E-LRP

In this chapter four models for the two-echelon location-routing problem are presented.

The first three models derive directly from the classical formulations proposed in Toth and Vigo ([96], [97]) for the VRP. The last formulation, instead, is based on a multi-depot vehicle-routing formulation ($MDVRP$) proposed by Dondo and Gerdá [37], which uses assignment and sequencing variables.

The chapter concludes with several computational results on small, medium and large instances obtained with a commercial solver.

4.1 2E-LRP Setting

Before presenting the formulations, the general setting of the problem is given.

- Sets:

$P = 1, \dots, P$	set of the possible platform locations
$S = 1, \dots, s$	set of the possible satellite locations
$Z = 1, \dots, z$	set of customer
$G = 1, \dots, g$	set of first echelon vehicles, *urban trucks*
$V = 1, \dots, v$	set of second echelon vehicle, *city freighters*

- Parameters:

H_p	fixed cost for opening a platform i, $p \in P$;
H_s	fixed cost for opening a satellite s, $s \in S$;

TCG	fixed cost for using a urban truck g, $g \in G$
TCV	fixed cost for using a city freighter v, $v \in V$
CTA_{ij}	transportation cost on the first echelon from a node i and j, $i, j \in P \cup S$;
CTB_{ij}	transportation cost on the second echelon from node i and j, $i, j \in S \cup Z$;
K_p	capacity of platform i, $p \in P$;
K_s	capacity of satellite s, $s \in S$;
UG	capacity of urban trucks g, $g \in G$;
UV	capacity of city freighters v, $v \in V$;
D_z	demand of each client, $z \in Z$.

4.2 A three-index 2E-LRP formulation

This model is an adaptation of the multi-echelon LRP formulation proposed by Ambrosino and Scutellá [2], obtained extending the single-echelon LRP formulation of Pearl and Daskin [84].

Given the previous problem setting, the three index formulation is based on the following sets of variables:

$r_{ij}^{g} = \{0, 1\}$	*1*, if i precedes j in the routing of the first echelon, performed by urban truck g, *0* otherwise
$x_{ij}^{v} = \{0, 1\}$	*1*, if i precedes j in the routing of the second echelon, performed by city freighter v, *0* otherwise
$w_{sz} = \{0, 1\}$	*1*, if the customer z, $z \in Z$, is assigned to satellite s, $s \in S$, *0* otherwise
$y_p = \{0, 1\}$	*1*, if a platform is opened at node i, $p \in P$, *0* otherwise
$y_s = \{0, 1\}$	*1*, if a platform is opened at node s, $s \in S$, *0* otherwise
$t^g = \{0, 1\}$	*1*, if urban truck g is used, $g \in G$, *0* otherwise
$t^v = \{0, 1\}$	*1*, if city freighter v is used, $v \in V$, *0* otherwise
$f_{ps}^{g} \geq 0$	is the quantity of good transported by the platform p, $p \in P$, to the satellite s, $s \in S$, with urban truck g, $g \in G$.

The problem can be formulated as follows:

$$\text{Minimize} \quad \sum_{p \in P} H_p\ y_p + \sum_{s \in S} H_s\ y_s + \sum_{g \in G} TCG\ t^g + \sum_{v \in V} TCV\ t^v +$$
$$+ \sum_{v \in V} \sum_{i \in S \cup Z} \sum_{j \in S \cup Z} CTB_{ij}\ x_{ij}^v + \sum_{g \in G} \sum_{p \in P \cup S} \sum_{j \in P \cup S} CTA_{ij}\ r_{ij}^g \tag{4.1}$$

subject to

$$\sum_{v \in V} \sum_{j \in S \cup Z} x_{zj}^v = 1 \qquad \forall z \in Z \tag{4.2}$$

$$\sum_{l \in S \cup Z} x_{lj}^v - \sum_{l \in S \cup Z} x_{jl}^v = 0 \qquad \forall j \in Z \cup S, \forall v \in V \tag{4.3}$$

$$\sum_{l \in \Omega} \sum_{h \in \overline{\Omega}} \sum_{v \in V} x_{lh}^v \geq 1 \qquad \forall \Omega \subset S \cup Z, with\ S \subseteq \Omega \tag{4.4}$$

$$\sum_{l \in S \cup Z} \sum_{j \in S} x_{lj}^v \leq 1 \qquad \forall v \in V \tag{4.5}$$

$$\sum_{g \in G} \sum_{j \in P \cup S} r_{lj}^g = y_l \qquad \forall l \in S \tag{4.6}$$

$$\sum_{l \in P \cup S} r_{lh}^g - \sum_{l \in P \cup S} r_{hl}^g = 0 \qquad \forall h \in P \cup S, \forall g \in G \tag{4.7}$$

$$\sum_{l \in \Omega} \sum_{h \in \overline{\Omega}} \sum_{g \in G} r_{lh}^g \geq y_j \qquad \forall j \in S, \forall \Omega \subset P \cup S, with\ P \subseteq \Omega, \overline{\Omega} \cap \{j\} \neq \oslash \tag{4.8}$$

$$\sum_{l \in P \cup S} \sum_{j \in P} r_{lj}^g \leq 1 \qquad \forall g \in G \tag{4.9}$$

$$\sum_{h \in S \cup Z} x_{zh}^v + \sum_{h \in S \cup Z} x_{sh}^v - w_{sz} \leq 1 \qquad \forall z \in Z, \forall v \in V, \forall s \in S \tag{4.10}$$

$$\sum_{s \in S} w_{sz} = 1 \qquad \forall z \in Z \tag{4.11}$$

$$\sum_{p\in P}\sum_{g\in G} f_{ps}^{g} - \sum_{z\in Z} D_z\ w_{sz} = 0 \qquad \forall s \in S \tag{4.12}$$

$$\sum_{s\in S} f_{ps} - K_p\ y_p \leq 0 \qquad \forall p \in P \tag{4.13}$$

$$\sum_{p\in P}\sum_{g\in G} f_{ps}^{g} - K_s\ y_s \leq 0 \qquad \forall s \in S \tag{4.14}$$

$$UG \sum_{h\in S\cup P} r_{sh}^{g} - f_{ps}^{g} \geq 0 \qquad \forall g \in G, \forall s \in S, \forall p \in P \tag{4.15}$$

$$UG \sum_{h\in S\cup P} r_{ph}^{g} - f_{ps}^{g} \geq 0 \qquad \forall g \in G, \forall s \in S, \forall p \in P \tag{4.16}$$

$$\sum_{i\in Z} D_z \sum_{j\in S\cup Z} x_{zj}^{v} \leq UV\ t^{v} \qquad \forall v \in V \tag{4.17}$$

$$\sum_{p\in P}\sum_{s\in S} f_{ps}^{g} \leq UG\ t^{g} \qquad \forall g \in G \tag{4.18}$$

$$\begin{aligned}
r_{ij}^{g} &= \{0,1\} && \forall i,j \in P \cup S, g \in G \\
x_{ij}^{v} &= \{0,1\} && \forall i,j \in S \cup Z, v \in V \\
w_{sz} &= \{0,1\} && \forall s \in S, z \in Z \\
y_p &= \{0,1\} && \forall p \in P \\
y_s &= \{0,1\} && \forall s \in S \\
t^{g} &= \{0,1\} && \forall g \in G \\
t^{v} &= \{0,1\} && \forall v \in V \\
f_{ps}^{g} &\geq 0 && \forall p \in P, s \in S, g \in G
\end{aligned} \tag{4.19}$$

The objective function 4.1 is the sum of six cost components: location cost for platforms, location cost for satellites, fixed cost for usage of urban trucks, fixed cost for usage of city freighters, transportation cost on the second and on the first echelons.

For what concerns the constraints of the model, they can be classified in function of routing on first and second echelon, flow conservation and capacity constraints, consistency constraints.
Routing constraints for the second echelon.

- Constraints (4.2) impose that each customer $z \in Z$ has just one leaving arc, i.e. it is served by exactly one city freighter v.
- Constraints (4.3) impose that the number of arcs for each vehicle, $v \in V$ entering in a node, $i \in Z \cup S$ is equal to the number of arcs leaving the node (customer or satellite), i.e. each used truck entering in a node has also to leave the same node.
- Constraints (4.4) are subtour elimination constraints. They impose the presence of at least a satellite in each route performed by a city freighter. These constraints can be replaced by other subtour elimination constraints proposed for the classical VRP problem.
- Constraints (4.5) impose that each city freighter, $v \in V$, has to be assigned unambiguously to one satellite, $s \in S$, i.e. each vehicle can perform just one route.

Routing constraints for the first echelon.

- Constraints (4.6), (4.7), (4.8), (4.2) impose the same conditions defined for the routing of the second echelon, but for the variables involved in the first echelon, i.e. for the routing between platforms and satellites with urban trucks.

Flow conservation and capacity constraints for facilities and vehicles

- Constraints (4.12) are the flow conservation constraints at the satellites, i.e. the amount of flow leaving the platforms is to be equal to the total demand of the customers. No storing is allowed at the satellites.
- Constraints (4.13) impose that the flow leaving a platform $p \in P$ has to be less than its own capacity, if the facility is open.
- Constraints (4.14) impose that the flow entering in a satellite $s \in S$ has to be less than its own capacity, if the facility is open.

- Constraints (4.17) impose that the demand assigned to a city freighter $v \in V$ has to be less than its own capacity, if the vehicle is used.

- Constraints (4.18) impose that the amount of flow transferred by a urban truck $g \in G$ has to be less than its own capacity, if the vehicle is used.

Consistency constraints between routing, allocation and flow variables.

- Constraints (4.10) link the allocation and routing components. In fact by constraint 4.2 each customer is assigned to exactly one route v. This constraint together with constraints 4.3, 4.4 and 4.5, imply that there must be exactly one satellite on the route of truck v. Therefore if we consider any given customer z^*, assigned to a route v^*, which contains also a satellite s^*, then we have that $\sum_{h \in S \cup Z} x^{v^*}_{z^* h} = 1$ and $\sum_{h \in S \cup Z} x^{v^*}_{s^* h} = 1$, and consequently $w_{z^* s^*} = 1$. Therefore the client z^* is assigned to the satellite s^*. If customer is not on a route starting from satellite s^*, then constraints 4.10 are satisfied for both $w_{sz} = 0$ and $w_{sz} = 1$, but since each customer has to be assigned to just one satellite, then it will be assigned to the one satisfying its demand.

- Constraints (4.11) is a redundant constraint which imposes that each customer z has to be assigned to a satellite s. This constraint allows anyway to slightly improve the bound of LP relaxation of the problem.

- Constraints (4.15) and (4.16) guarantee that the flow on a vehicle g from a platform p to a satellite s, $f^g_{ps} \geq 0$ if and only if both the satellite and the platform are visited by the same vehicle g.

In end constraints (4.19) express the integrality constraints for the binary variables involved in the formulation and non-negativity constraints for the flow variables.

In literature several papers use the three-index formulation for the single-echelon location routing problem. Even if the basic structure is the same, they present differences for what concern the subtour elimination constraints. In fact in Tuzun and Burke [99], Wu et al. [106] and Wang et al.[101], the subtour elimination constraints proposed by Miller et al. [76] and adapted to the $MDVRP$ are used. Therefore subtour elimination constraints can be expressed as:

$$L_i - L_j + (S+Z)\sum_{v\in V} x_{ij}^v \leq (S+Z-1) \qquad \forall i,j \in Z\cup S, i\neq j \tag{4.20}$$

$$L_i - L_j + (P+S)\sum_{g\in G} r_{ij}^g \leq (P+S-1) \qquad \forall i,j \in S\cup P, i\neq j \tag{4.21}$$

where L_i and L_j are continuous non-negative variables.

This constraint can be also specialized for each customer on the second echelon and for each satellite on the first echelon with the following expression:

$$L_{i,v} - L_{j,v} + Z\cdot x_{ij}^v \leq (Z-1) \qquad \forall i,j \in Z, i\neq j, v\in V \tag{4.22}$$

$$L_{i,g} - L_{j,g} + S\cdot r_{ij}^g \leq (S-1) \qquad \forall i,j \in S, i\neq j, g\in G \tag{4.23}$$

where $L_{i,v}$, $L_{j,v}$, $L_{i,g}$ and $L_{j,g}$ are continuous non-negative variables.

Constraints (4.20) and (4.21) will be used in the proceeding of the work. Anyway, it is important to underline that the usage of these constraints provide worse lower bounds than subtour elimination expressed by constraints (4.4) and (4.8).

4.3 A two-index 2E-LRP formulation

This formulation is an extension of the two-index formulation for the single-echelon location-routing problem proposed in Prins et al. [85]. In this formulation a variable is associated to each arc of the network. Let us consider a network $G(N,A)$. Two sets of arcs can be defined, referred as A and B. Set A is composed by all the arcs connecting the elements of the first echelon, i.e. the arcs connecting two satellites or a satellite and a platform, whereas set B is composed by all the arcs connecting elements of the second echelon, i.e. two customers or a customer and a satellite. At this point the following arc variables are defined:

$r_a^g = \{0, 1\}$	*1* if arc a, $a \in A1$, is used in the route performed by urban truck in the first echelon g, *0* otherwise
$x_a^v = \{0, 1\}$	*1*, if arc a, $a \in A2$, is used in the route performed by city freighter in the second echelon v, *0* otherwise

Moreover if Ω is a subset of N, $\delta^+(\Omega)$ ($\delta^-(\Omega)$) is the set of arcs leaving (entering) Ω and $L(\Omega)$ the set of arcs with both extremities in Ω. In case a set is composed of a single node, the notation $\delta^+(j)$ ($\delta^-(j)$) will be used. Moreover let us consider two subsets Θ and Ω. The arcs entering in subset Θ with the origin in the subset Ω will be indicated as $\delta^-(\Theta : \Omega)$. In the opposite case the notation is $\delta^+(\Theta : \Omega)$. Also in this case a single representative product is present at the opened facilities and the following flow variables are defined:

$f_a^g \geq 0$	continuous flow variable that represents the flow transported on the arc a, $a \in A$ from the platform p, $p \in P$, to the satellite s, $s \in S$, with urban truck g, $g \in G$.

Given the setting of *Section 4.1* the problem can be modeled as follows:

$$\text{Minimize} \quad \sum_{i \in I} H_i\ y_i + \sum_{s \in S} H_s\ y_s + \sum_{g \in G} TCG\ t^g + \sum_{v \in V} TCV\ t^v + \\ + \sum_{v \in V} \sum_{a \in B} CTB_a\ x_a^v + \sum_{g \in G} \sum_{a \in A} CTA_a\ r_a^g \tag{4.24}$$

Subject to

$$\sum_{v \in V} \sum_{a \in \delta^-(z)} x_a^v = 1 \qquad \forall z \in Z \tag{4.25}$$

$$\sum_{a \in \delta^+(i)} x_a^v - \sum_{a \in \delta^-(i)} x_a^v = 0 \qquad \forall i \in S \cup Z, \forall v \in V \tag{4.26}$$

$$\sum_{a \in \delta^+(S)} x_a^v \leq 1 \qquad \forall v \in V \tag{4.27}$$

$$\sum_{a \in L(\Omega)} x_a^v \leq |\Omega| - 1 \qquad \forall \Omega \subseteq J, \forall v \in V \tag{4.28}$$

$$\sum_{g \in G} \sum_{a \in \delta^-(s)} r_a^g = y_s \qquad \forall s \in S \tag{4.29}$$

$$\sum_{a \in \delta^+(i)} r_a^g - \sum_{a \in \delta^-(i)} r_a^g = 0 \qquad \forall p \in P \cup S, \forall g \in G \tag{4.30}$$

$$\sum_{a \in \delta^+(P)} r_a^g \leq 1 \qquad \forall g \in G \tag{4.31}$$

$$\sum_{a \in L(\Omega)} r_a^v \leq |\Omega| - 1 \qquad \forall \Omega \subseteq S, \forall g \in G \tag{4.32}$$

$$\sum_{s \in S} w_{sz} = 1 \qquad \forall z \in Z \tag{4.33}$$

$$\sum_{a \in \delta^+(i) \cap \delta^-(Z)} x_a^v - \sum_{a \in \delta^-(j)} x_a^v \leq 1 + w_{ij} \qquad \forall i \in S, \forall j \in Z, \forall v \in V \tag{4.34}$$

$$UG \sum_{a \in \delta^+(s)} r_a^g - f^g_{a_{|a \in \delta^-(s:p)}} \geq 0 \qquad \forall s \in S, p \in P, g \in G \tag{4.35}$$

$$UG \sum_{a \in \delta^+(p)} r_a^g - f^g_{a_{\|a \in \delta^-(s:p)}} \geq 0 \qquad \forall s \in S, p \in P, g \in G, \tag{4.36}$$

$$\sum_{a \in \delta^-(s:P)} \sum_{g \in G} f_a^g - \sum_{i \in Z} D_i \; w_{is} = 0 \qquad \forall s \in S \tag{4.37}$$

$$\sum_{a \in \delta^-(j):P} \sum_{g \in G} f_a^g - K_s \; y_s \leq 0 \qquad \forall s \in S \tag{4.38}$$

$$\sum_{a \in \delta^+(i)} f_a^g \leq K_p \; y_P \qquad \forall p \in P \tag{4.39}$$

$$\sum_{i \in Z} \sum_{a \in \delta^-(i)} D_i \; x_a^v \leq UV \; t^v \qquad \forall v \in V \tag{4.40}$$

$$\sum_{a \in \delta^+(P)} f_a^g \leq UG\, t^g \qquad \forall g \in G \tag{4.41}$$

$$\begin{aligned}
r_a^g &= \{0,1\} && \forall a \in A, g \in G\\
x_a^v &= \{0,1\} && \forall a \in B, v \in V\\
w_{sz} &= \{0,1\} && \forall s \in S, z \in Z\\
y_p &= \{0,1\} && \forall p \in P\\
y_s &= \{0,1\} && \forall s \in S\\
t^g &= \{0,1\} && \forall g \in G\\
t^v &= \{0,1\} && \forall v \in V\\
f_{ps}^g &\geq 0 && \forall p \in P, s \in S, g \in G
\end{aligned} \tag{4.42}$$

Also in this model the objective function (4.24) is expressed as the sum of six components: location costs for satellites and platforms, cost for the usage of first and second echelon vehicles and routing costs. Constraints follow the same structure of the three-index model. A brief explanation follows.

Routing constraints for second echelon.

- Constraints (4.25) impose that each customer z, $z \in Z$, has to be served by just one vehicle v, $v \in V$.
- Constraints (4.26) impose that for each vehicle v, $v \in V$, the number of arcs entering a node is equal to the number of arcs leaving the node, for i, $i \in S \cup Z$
- Constraints (4.27) impose that each vehicle v, $v \in V$, can ba assigned to no more than one satellite s, $s \in S$.
- Constraints (4.28) are subtour elimination constraints for the second level routes.

Routing constraints for first echelon.

- Constraints (4.29) impose that each open satellite s, $s \in S$, has to be served by a first echelon vehicle g, $g \in G$.
- Constraints (4.30) impose that for each vehicle g, $g \in G$, the number of arcs entering a node is equal to the number of arcs leaving the node, for i, $i \in P \cup S$

- Constraints (4.31) impose that each first echelon vehicle g, $g \in G$ has to be assigned to one platform p, $p \in P$.
- Constraints (4.32) impose the subtour elimination constraints for the first echelon.

Flow conservation and capacity constraints for facilities and vehicles.

- Constraints (4.37) are the flow balance constraints at the satellites s, $s \in S$.
- Constraints (4.38) and (4.39) are the capacity constraints for the two kinds of facilities.
- Constraints (4.41) and (4.40) impose the capacity constraints respectively for each first and second echelon vehicle.

Consistency constraints between routing, allocation and flow variables.

- Constraints (4.34) impose that if a satellite and a customer are served by the same vehicle, then the customer is assigned to that satellite.
- Constraints (4.33) impose that each client has to be assigned to a satellite.
- Constraints (4.35) and (4.36) are consistency constraints between flow and routing variables on the first echelon

In end constraints (4.42) express the integrality constraints for the binary variables involved in the formulation and non-negativity constraints for the flow variables.

4.4 A one-index 2E-LRP formulation

The last formulation derived from VRP is an extension of the one-index formulation proposed for the single-echelon location routing problem. In this formulation, basically, a variable is defined for all the possible routes, or just for the ones respecting a predefined criterion on maximum length, or time consideration, etc.. This is often referred as set-partitioning formulation and it uses an exponential number of binary variables for each feasible route. In our case two different sets of routes have to be defined, one for the first echelon $A = 1, 2, \ldots, a$ and one for the second echelon $B_i = 1, 2, \ldots, b$. Let us also indicate as A_p, the subset of A composed of the routes starting from the platform p. Each route is associated to a binary variable, respectively r_i for the first echelon and x_i for the second echelon, which assume value 1 if a route is used, 0 otherwise. Moreover for each first-echelon route a flow variable $f(i)$, $i \in A$ is defined. Each path is also associated to a cost, indicated with CTA_i for the first-echelon path and with CTB_i for the second-echelon. For each echelon a path-node incidence matrix, respectively $\boldsymbol{CA}$ and $\boldsymbol{CB}$, is defined, whose generic elements assume the following values:

$ca_{is} = \{0, 1\}$	*1*, if a satellite s, $s \in S$ is covered by the path i, $i \in A$, 0 otherwise
$cb_{iz} = \{0, 1\}$	*1*, if a customer z, $z \in Z$ is covered by the path i, $i \in B$, 0 otherwise

Other two incidence matrices have still to be defined. The matrix $\boldsymbol{EA}$, of dimensions $(P \times S)$ and $\boldsymbol{EB}$ of dimensions $(S \times Z)$, whose generic elements assume the following values:

$ea_{ps} = \{0, 1\}$	*1*, if a satellite s, $s \in S$ is covered by a platform p, $p \in P$, 0 otherwise
$eb_{sz} = \{0, 1\}$	*1*, if a customer z, $z \in Z$ is covered by a satellite s, $s \in S$, 0 otherwise

Therefore the problem can be formulated as follows:

$$\text{Minimize} \quad \sum_{p \in P} H_p\ y_p + \sum_{s \in S} H_s\ y_s + TCG \sum_{i \in A} r_i + TCV \sum_{i \in B} s_i + \\ + \sum_{i \in A} CTA_i\ r_i + \sum_{i \in B} CTB_i\ x_i \tag{4.43}$$

subject to

$$\sum_{s \in S} eb_{sz}\ y_s = 1 \qquad \forall z \in Z \tag{4.44}$$

$$\sum_{i \in B} cb_{iz}\ x_i = \sum_{j \in S} eb_{jz}\ y_j \qquad \forall z \in Z \tag{4.45}$$

$$\sum_{p \in P} ea_{ps}\ y_p = y_s \qquad \forall j \in S \tag{4.46}$$

$$\sum_{i \in A} a_{is}\ r_i = \sum_{p \in P} ea_{ps}\ y_p \qquad \forall s \in S \tag{4.47}$$

$$\sum_{i \in A} f_i\ a_{is} - \sum_{z \in Z} D_z\ eb_{sz}\ y_s = 0 \qquad \forall p \in P \tag{4.48}$$

$$\sum_{z \in Z} D_z\ eb_{sz}\ y_s \leq K_s\ y_s \qquad \forall s \in S \tag{4.49}$$

$$\sum_{i \in A_p} f_i \leq K_p\ y_p \qquad \forall p \in P \tag{4.50}$$

$$UG\ r_i - f_i \geq 0 \qquad \forall i \in A \tag{4.51}$$

$$\begin{aligned} r_i &= \{0,1\} && \forall i \in A \\ x_i &= \{0,1\} && \forall i \in B \\ y_p &= \{0,1\} && \forall p \in P \\ y_s &= \{0,1\} && \forall s \in S \\ f_i &\geq 0 && \forall i \in A \end{aligned} \tag{4.52}$$

The explanation of the model objective function and constraints can be straightforward obtained by the discussion provided for the previous models. Anyway it can be summarized as follows:

- Objective function (4.43) minimize the overall costs: location costs, transportation costs and vehicle costs.
- Constraints (4.44) impose that each customer is served by just one open satellite.
- Constraint (4.45) imposes that if a customer is served by a satellite, then it has to be served on just one route passing through the same satellite and on the other side if a customer is served on a route starting from a satellite, then the customer is served by same satellite.

- Constraint (4.46) and (4.47) impose the same routing conditions described for the customers, but referred to the open satellites and platforms.

- Constraints (4.48) are flow balance constraints for satellites.

- Constraints (4.49) and (4.50) are instead the capacity constraints respectively for the satellites and platforms.

- Constraints (4.51) are consistency constraints between flow and routing variables.

- Constraints (4.51) are finally the binary and non-negativity constraints for the variables.

4.5 Assignment-based 2E-LRP formulation

Another formulation has been realized for the *2E-LRP* which comes from an adaptation of the MDVRP formulation, proposed by Dondo and Cerdá [37]. In this formulation they define a multi-depot vehicle routing problem using just two-index variables, more precisely assignment variables and sequencing variables. For this reason it is referred as *assignment based formulation*. It requires the definition of different integer and non negative variables for the two echelons:

$a_{zv} = \{0, 1\}$	assignment variable for the second echelon which assumes value 1 if a customer z, $z \in Z$, is assigned to a city freighter v, $v \in V$, 0 otherwise;
$b_{sv} = \{0, 1\}$	assignment variable for the second echelon which assumes value 1 if a city freighter v, $v \in V$, is assigned to a satellite s, $s \in S$, 0 otherwise;
$w_{sz} = \{0, 1\}$	assignment variable for the second echelon which assume value 1 of a customer z, $z \in Z$, is assigned to a satellite s, $s \in S$;

$x_{ij} = \{0, 1\}$ sequencing variable to denote that customer i is visited before customer j, ($x_{ij} = 1$), or after ($x_{ij} = 0$), just in case they are both serviced by the same city freighter ($a_{iv} = a_{jv}$), otherwise the value of x_{ij} will be meaningless. It is important to underline that it is defined just a single variable x_{ij} for each pair of nodes (i, j) that can share the same tour. Therefore the relative ordering of nodes (i, j) is established by the variable x_{ij} such that $ord(i) < ord(j)$ where $ord(i)$ indicates the relative position of the element i in the customer set Z. In this way the number of sequencing variable is cut by half;

$m_{sg} = \{0, 1\}$ assignment variable for the first echelon which assumes value 1 if the open satellite s, $s \in S$ is assigned to the urban truck $g \in G$, 0 otherwise;

$n_{pg} = \{0, 1\}$ assignment variable for the first echelon which assume value 1 if a urban truck g, $g \in G$ is assigned to an open platform p, $p \in P$, 0 otherwise;

$r_{ij} = \{0, 1\}$ sequencing variable to denote that satellite i is visited before customer j, ($r_{ij} = 1$), or after ($r_{ij} = 0$), just in case they are both serviced by the same urban truck ($m_{iv} = m_{jv}$). Otherwise the value of r_{ij} will be meaningless. Also in this case it is important to underline the fact that the number of variables is cut by half, since a single variable r_{ij} is considered for each couple of nodes;

$CB(z) \geq 0$ variable indicating the accumulated routing cost on the second echelon up to a customer z, $z \in Z$;

$CA(s) \geq 0$ variable indicating the accumulated routing cost on the first echelon up to a satellite s, $s \in S$;

$CV(v) \geq 0$ variable indicating the total routing cost for a city freighter v, $v \in V$;

$CG(g) \geq 0$ variable indicating the total routing cost for a urban truck g, $g \in G$;

Maintaining the setting of *Section 4.1*, the problem can be modeled as follows:

$$\text{Minimize} \quad \sum_{p\in P} H_p\ y_p + \sum_{s\in S} H_s\ y_s + TCG \sum_{p\in P} \sum_{g\in G} n_{pg} + \\ + TCV \sum_{s\in S} \sum_{v\in V} b_{sv} + \sum_{v\in V} CV(v) + \sum_{g\in G} CG(g) \qquad (4.53)$$

Subject to

$$\sum_{v\in V} a_{zv} = 1 \qquad \forall z \in Z \qquad (4.54)$$

$$\sum_{s\in S} b_{sv} \leq 1 \qquad \forall v \in V \qquad (4.55)$$

$$CB_i \geq CTB_{si}\,(b_{jv} + a_{iv} - 1) \qquad \forall i \in Z, s \in S, v \in V \qquad (4.56)$$

$$CB_j \geq C_i + CTB_{ji} - M_c\,(1 - x_{ij}) - M_c\,(2 - a_{jv} + a_{iv}) \qquad \forall i,j \in Z_{i<j}, v \in V \qquad (4.57)$$

$$CB_i \geq C_j + CTB_{ij} - M_c\,(x_{ij}) - M_c\,(2 - a_{jv} + a_{iv}) \qquad \forall i,j \in Z_{i<j}, v \in V \qquad (4.58)$$

$$CV_v \geq C_i + CTB_{is} - M_c\,(2 - b_{sv} - a_{iv}) \qquad \forall i \in Z, s \in S, v \in V \qquad (4.59)$$

$$\sum_{g\in G} m_{sg} = y_s \qquad \forall s \in S \qquad (4.60)$$

$$\sum_{j\in I} n_{jg} \leq 1 \qquad \forall g \in G \qquad (4.61)$$

$$CA_i \geq CTA_{ji}\,(n_{jg} + a_{ig} - 1) \qquad \forall i \in S, j \in P, g \in G \qquad (4.62)$$

$$CA_j \geq CA_i + CTA_{ji} - M_c\,(1 - r_{ij}) - M_c\,(2 - a_{jg} + a_{ig}) \qquad \forall i,j \in S_{i<j}, g \in G \qquad (4.63)$$

$$CA_i \geq CA_j + CTA_{ij} - M_c\,(r_{ij}) - M_c\,(2 - a_{jg} + a_{ig}) \qquad \forall i,j \in S_{i<j}, g \in G \tag{4.64}$$

$$CG_g \geq C_i + CTA_{ip} - M_c\,(2 - b_{pg} - a_{ig}) \qquad \forall i \in S, p \in P, g \in G \tag{4.65}$$

$$\sum_{p\in P}\sum_{g\in G} f_{ps}^g - \sum_{z\in Z} D_z\ w_{sz} = 0 \qquad \forall s \in S \tag{4.66}$$

$$\sum_{z\in Z} D_z\ w_{sz} \leq K_s\ y_s \qquad \forall s \in S \tag{4.67}$$

$$\sum_{s\in S} f_{ps}^g - K_p\ y_p \leq 0 \qquad \forall p \in P \tag{4.68}$$

$$\sum_{i\in Z} d_i\ a_{iv} \leq UV \sum_{s\in S} b_{sv} \qquad \forall v \in V \tag{4.69}$$

$$a_{zv} + b_{sv} - w_{sz} \leq 1 \qquad \forall i \in Z, s \in S, v \in V \tag{4.70}$$

$$UG\ n_p^g - f_{ps}^g \geq 0 \qquad \forall g \in G, \forall s \in S, \forall p \in P \tag{4.71}$$

$$UG\ m_s^g - f_{ps}^g \geq 0 \qquad \forall g \in G, \forall s \in S, \forall p \in P \tag{4.72}$$

$$\begin{array}{rlrl}
a_{zv} = \{0,1\} & \forall z \in Z, v \in V & b_{sv} = \{0,1\} & \forall s \in S, v \in V \\
w_{sz} = \{0,1\} & \forall z \in Z, s \in S & x_{ij} = \{0,1\} & \forall i,j \in Z_{(i<j)} \\
m_{sg} = \{0,1\} & \forall s \in S, g \in G & n_{pg} = \{0,1\} & \forall p \in P, g \in G \\
r_{ij} = \{0,1\} & \forall i,j \in S_{(i<j)} & y_s = \{0,1\} & \forall s \in S \\
y_p = \{0,1\} & \forall p \in P & & \\
CB_z \geq 0 & \forall z \in Z & CA_s \geq 0 & \forall s \in S \\
CV_v \geq 0 & \forall v \in V & CG_g \geq 0 & \forall g \in G
\end{array} \tag{4.73}$$

The objective function (4.53), as in previous models, is aimed at minimizing the total cost. For what concerns the constraints the same classification of the previous sections is used for their explanation:

Routing constraints for the second echelon.

- Constraints (4.54) assign customers to urban trucks. Every customer node $z \in Z$ must be serviced by a single vehicle $v \in V$ in a single source way.
- Constraints (4.55) assign satellites to urban trucks. Every used vehicle $v \in V$ should be allocated to a single open satellite $s \in S$ to which it returns after visiting all the assigned customers. The required fleet size is a problem variable to be determined simultaneously with the best set of routes.
- Constraints (4.56) defines the least cost for a urban truck to reach a customer. The cost of traveling from a satellite $s \in S$ to a node i, referred as (CB_i) must be greater than or equal to CTB_{si} only if the node $i \in Z$ is assigned to a vehicle $v \in V$ $(Y_{iv} = 1)$ starting his route from the depot s, $X_{sv} = 1$. This is so because, since CTB_{si} is the least travel cost from a depot s to node i. They become binding just in case customer i is the first visited by vehicle v.
- Constraints (4.57) and (4.58) define the relationship between traveling costs up to nodes $i, j \in Z$ on the same tour. In fact being CTB_{ij} the least travel cost from node i to node j on the vehicle v, if both nodes are on the same tour, i.e. $(Y_{iv} = Y_{jv} = 1$, for a vehicle v and node i is visited before, i.e. $(S_{ij} = 1)$, then constraints (4.57 states that the routing cost from the depot to node j, (CB_j) must always be greater than (CB_i) by at least CTB_{ij}. On the other side if node j is visited earlier $(Sij = 0)$, the reverse statement holds. Constraints (4.57) and (4.58)can become redundant whenever nodes $i, j \in Z$ are serviced by different vehicles, i.e. $(Yiv + Yjv < 2)$ for any $v \in V$. By definition, MC is a large positive number.
- Constraints (4.59) define the routing cost for each urban truck $v \in V$. In fact the routing cost for a urban truck $v \in V$, referred as (CV_v) must always be greater than the routing cost from the satellite to any node i along the route (i.e. greater than (CB_i)) by at least the amount CTB_{is}. Indeed, the last node visited by vehicle v is the one finally defining the value of (CV_v).

Routing constraints for the first echelon.

- Constraints from (4.60), to (4.64) have the same effect of the routing constraints for the second echelon and therefore the explanation can be directly derived from the previous discussion.

Flow conservation and capacity constraints for facilities and vehicles.

- Constraints (4.66) are flow conservation constraints at the open satellite $s \in S$.

- Constraints (4.67), (4.68) and (4.69) are capacity constraints related to respectively satellites, platforms and urban trucks.

Consistency constraints between routing, allocation and flow variables.

- Constraints (4.70) are consistency constraints between assignment variables. If a satellite s and a customer z are assigned to the same vehicle v, then the customer i is assigned to the satellite s.

- Constraints (4.71) and (4.72) are consistency constraints between flow and assignment variables. A city freighter u can transport flow on the first echelon from a platform p to a satellite s if only if it is assigned to them. Moreover it imposes constraint on the maximum capacity for vehicle u.

Finally, as in the other models, constraints (4.73) are integrality and non-negativity constraints.

4.6 2E-LRP models computational results

In this section the experimental results for three-index formulation and assignment based formulation on three sets of 2E-LRP instances are presented. Models have been solved with the usage of a commercial solver, Xpress-MP, and instances were run on an Intel(R) Pentium(R) 4(2.40 GHz, RAM 4.00 GB).

Instances have been generated through an *instance generator* developed in *C++*, whose functioning mechanism is reported in appendix of the thesis. Here we just point out that the three sets of instances differ for the spatial distribution of satellites.

The used notation to describe an instance is the following: $Testset-PSZ$. Therefore *I1-51050* refers to an instance of set *I1* with *5* platforms, 10 *satellites* and *50* customers.

In tables (4.1, 4.2, 4.3) the results obtained on small instances with three-index (*3i*) and assignment based (*ab*) formulations will be reported. In particular for each instance and for each formulation the value of the best lower bound (*3i-LB*, *ab-LB*), the best determined solution (*3i-BS*, *ab-BS*) and the related computation time (*3i-CPU*, *ab-CPU*) in seconds are reported. In particular concerning CPU time, the execution has been limited to *7200* seconds.

	3-index-formulation			Assignment-based formulation		
Instance	3i-LB	3i-BS	3i-CPU	ab-LB	ab-BS	ab-CPU
I1-238	591.83	591.83	896.21	591.83	591.83	10.23
I1-239	878.69	878.69	1489.51	878.69	878.69	9.87
I1-248	625.96	625.96	1678.11	625.96	625.96	175.60
I1-2410	862.91	862.91	3097.30	862.91	862.91	582.90
I1-2415	1001.82	1105.67	3612.00	1105.67	1105.67	1469.90
I1-3510	829.25	829.25	4587.90	829.25	829.25	2194.70
I1-3515	989.18	1019.57	6118.00	1019.57	1019.57	3893.50
I1-2820	881.27	1129.37	7200.00	737.57	1055.65	7200.00
I1-2825	785.82	1086.43	7200.00	580.00	992.08	7200.00
I1-21015	714.69	732.48	7200.00	554.71	732.48	7200.00
I1-21020	752.03	1041.93	7200.00	595.00	951.01	7200.00
I1-21025	627.96	1170.72	7200.00	301.71	1334.94	7200.00
I1-3810	571.85	604.37	6219.21	604.37	604.37	4982.30
I1-3815	679.83	730.36	7200.00	515.00	730.36	7200.00
I1-3820	652.56	932.42	7200.00	351.00	898.75	7200.00
I1-3825	579.71	1141.26	7200.00	275.00	1224.12	7200.00
I1-31015	675.27	699.11	7200.00	435.00	796.33	7200.00
I1-31020	783.61	810.26	7200.00	398.02	917.02	7200.00
I1-31025	831.23	1291.68	7200.00	590.00	1316.75	7200.00
I1-41020	883.13	1397.81	7200.00	571.21	1208.72	7200.00
I1-41025	1045.17	1791.35	7200.00	715.75	1615.33	7200.00

Table 4.1: Results of 3-index and ab-based formulations on small instances I1.

From the previous tables we can observe that just for very small instances (up to 3 platforms, 5 satellites and 15 customers) it is possible to determine the optimal solution through the usage of a commercial solver within the predefined running time. For medium instances, instead, both formulations do not return the optimal solutions. The results provided by the two formulations are similar in terms of quality of solutions. On the other side the three-index formulation returns better bounds than the assignment-based formulation, which on its turn,

Instance	3-index-formulation			Assignment-based formulation		
	3i-LB	3i-BS	3i-CPU	ab-LB	ab-BS	ab-CPU
I2-238	589.38	589.38	1099.81	589.38	589.38	6.45
I2-239	413.54	413.54	1699.31	413.54	413.54	8.31
I2-248	605.40	605.40	2242.91	605.40	605.40	182.50
I2-2410	629.38	629.38	3251.40	629.38	629.38	834.30
I2-2415	912.73	912.73	3967.30	912.73	912.73	1525.30
I2-3510	551.45	551.45	4145.30	551.45	551.45	2281.50
I2-3515	1170.83	1170.83	6632.00	1170.83	1170.83	4365.50
I2-2820	663.94	858.07	7200.00	555.00	822.85	7200.00
I2-2825	776.49	947.84	7200.00	510.13	982.87	7200.00
I2-21015	706.12	727.72	7200.00	450.00	727.77	7200.00
I2-21020	672.63	879.15	7200.00	510.94	801.28	7200.00
I2-21025	665.10	1316.99	7200.00	362.62	1263.54	7200.00
I2-3810	495.08	504.20	7200.00	504.20	504.20	6412.23
I2-3815	655.37	685.48	7200.00	588.03	685.48	7200.00
I2-3820	691.77	805.38	7200.00	467.65	832.48	7200.00
I2-3825	813.16	1125.23	7200.00	490.00	1026.36	7200.00
I2-31015	657.82	812.13	7200.00	560.21	826.52	7200.00
I2-31020	732.32	817.05	7200.00	525.00	806.67	7200.00
I2-31025	833.97	1322.00	7200.00	628.55	1254.62	7200.00
I2-41020	751.62	1193.85	7200.00	582.13	1093.34	7200.00
I2-41025	983.23	1487.14	7200.00	700.86	1380.86	7200.00

Table 4.2: Results of 3-index and ab-based formulations on small instances I2.

Instance	3-index-formulation			Assignment-based formulation		
	3i-LB	3i-BS	3i-CPU	ab-LB	ab-BS	ab-CPU
I3-238	589.80	589.78	1215.12	589.80	589.78	8.13
I3-239	454.63	454.63	1099.23	454.63	454.63	7.10
I3-248	451.62	451.62	1562.45	451.62	451.62	164.70
I3-2410	546.36	546.36	2578.80	546.36	546.36	416.80
I3-2415	718.16	718.16	4167.30	718.16	718.16	1225.30
I3-3510	745.85	745.85	3845.30	745.85	745.85	2674.20
I3-3515	1033.79	1033.79	5948.50	1033.79	1033.79	3065.50
I3-2820	800.22	829.20	7200.00	755.00	829.20	7200.00
I3-2825	657.34	1100.31	7200.00	351.98	1229.31	7200.00
I3-21015	601.80	620.86	7200.00	507.68	620.86	7200.00
I3-21020	473.53	790.99	7200.00	240.00	796.11	7200.00
I3-21025	681.00	944.84	7200.00	318.33	1175.71	7200.00
I3-3810	412.91	412.91	5439.21	412.91	412.91	3376.23
I3-3815	605.76	624.55	7200.00	586.34	624.55	7200.00
I3-3820	652.49	707.57	7200.00	523.58	707.57	7200.00
I3-3825	665.76	1044.50	7200.00	383.03	977.10	7200.00
I3-31015	533.06	574.26	7200.00	372.83	574.26	7200.00
I3-31020	499.35	789.49	7200.00	318.44	830.79	7200.00
I3-31025	511.56	1119.47	7200.00	324.14	1038.58	7200.00
I3-41020	976.21	1393.62	7200.00	616.21	1287.23	7200.00
I3-41025	741.30	1191.40	7200.00	541.30	1089.40	7200.00

Table 4.3: Results of 3-index and ab-based formulations on small instances I3.

requires lower computation time on small instances.

Concerning instead medium and large instances (i.e. with more than 25 customers), the commercial solver does not provide good solution, neither good bounds with computation times of about *36000* seconds. For this reason we decided to solve medium and large instances decomposing the problem in a capacitated multi-level facility location problem and in two multi-depot vehicle routing problems, one for each echelon. The decomposition approach will be explained more in detail in the following chapter. Here we present the results obtained on three sets of medium and large size instances. The three subproblems have been solved with Xpress-MP solver to the optimum or until a predefined gap value between solver lower bound and best solution was reached. In tables 4.4, 4.5, 4.6 the solution of the decomposed approach, *DA-BS*, and the related computation time, *DA-CPU*, given by the sum of the computation time of each sub-problem (in seconds), are reported.

Instance	DA-BS	DA-CPU
I1-5850	1226.24	4421.30
I1-51050	1783.60	6134.90
I1-51075	1591.60	7512.60
I1-51575	1783.60	6134.90
I1-510100	2247.32	8033.80
I1-520100	2055.88	10218.10
I1-510150	2177.77	8407.10
I1-520150	1933.82	7786.60
I1-510200	2625.11	10119.50
I1-520200	3140.17	12750.30

Table 4.4: Results of decomposition approach on large-medium instances I1.

Instance	DA-BS	DA-CPU
I2-5850	1185.75	2023.34
I2-51050	1325.61	5039.50
I2-51075	1768.88	7061.00
I2-51575	1644.79	9499.40
I2-510100	2391.17	10379.60
I2-520100	2051.39	12405.60
I2-510150	2111.97	14060.90
I2-520150	1800.89	10134.50
I2-510200	2430.93	8871.80
I2-520200	2274.29	15602.10

Table 4.5: Results of decomposition approach on large-medium instances I2.

Instance	DA-BS	DA-CPU
I3-5850	1298.89	7741.90
I3-51050	1256.68	4929.60
I3-51075	1879.56	8720.00
I3-51575	1704.65	10903.90
I3-510100	2601.44	9199.60
I3-520100	2261.36	10724.50
I3-510150	1470.77	4243.90
I3-520150	1508.07	12240.50
I3-510200	2193.32	10045.10
I3-520200	2784.47	13319.40

Table 4.6: Results of decomposition approach on large-medium instances I3.

From tables 4.4, 4.5, 4.6, we observe that the decomposition approach requires high computation times, but on the other side it allows to determine good upper bounds, which cannot be obtained with previous models within reasonable computation times. In fact, for instance, the best solution obtained with three-index and ab-formulations for *I1-51050* in *36000* seconds is equal to *2370*, versus *1783* obtained with decomposition approach in *6134.90* seconds.

The shown results motivate the need to use a heuristic approach to solve the 2E-LRP. The following chapter is devoted to the description of a Tabu Search heuristic for the 2E-LRP and the previous results will be used as bounds to compare the goodness of the heuristic solutions in terms of quality of solution and computation times.

Chapter 5

Tabu Search heuristic for 2E-LRP

In this chapter a *Tabu Search* approach for the two-echelon location-routing problem is presented. The decomposition-based solution approaches proposed in literature are presented. Then the main issues of the proposed metaheuristic are described: decomposition of the problem, initial solution and evaluation criterion; neighborhood of a solution and related tabu settings; combination of subproblems solutions; stopping and diversification criteria.

5.1 Solution approaches

The heuristic approaches present in literature for the location-routing problem are based on the decomposition in its two components, i.e. location-allocation and problem. In some cases the problem is decomposed in three components, because the location and the allocation problems are treated separately. Then the sub-problems solutions are combined to obtain a solution for the whole problem. In literature four basic approaches can be distinguished [80], differing for the way they interrelate and solve each component:

1. *Sequential approach*: there is a hierarchical relation between the two problems. At first location problem is solved approximating routing costs with an estimation parameter. Then the routing problem is solved for the selected facilities.

2. *Iterative approach*: the two components of the problem are considered as equal. Therefore the two problems are solved iteratively exchanging information at each iteration, until a stopping criterion is verified.

3. *Nested approach*: the two components are not considered as on an equal. Therefore it is recognized a hierarchical structure of the general problem, which is considered basically as a location problem, where routing aspects are taken into account. The difference with the sequential approach is in the fact that in this case the routing problem is solved for each possible location solution.

4. *Clustering approach*: an assignment of customers to depots is obtained performing clustering operation. Then a capacitated vehicle routing problem for each depot and for the assigned clusters is solved.

Obviously, for a two-echelon location-routing problem, the possible combinations of the components increase since we have two location and two routing sub-problems. Therefore the previous approaches have to be adapted as follows for the 2E-LRP:

1. *Sequential approach*: a multi-level capacitated facility location problem is solved by exact or heuristic methods and the two multi-depot vehicle routing sub-problems are solved sequentially starting from the second echelon. This is the approach used in *Section 4.6* for medium and large size instances.

2. *Iterative approach*: the four components of the problem are considered as equal. Therefore the two location and the two routing problems are solved iteratively exchanging information at each iteration, until a stopping criterion is verified. In this case the fundamental issue is the definition of the mechanism to exchange the information.

3. *Nested approach*: a multi-level capacitated facility location problem is solved more times, obtaining different solutions. Then the two routing sub-problems are solved sequentially for each different location solution on the two echelons.

4. *Clustering approach*: two possibilities can be defined. One foresees a first clustering on the first echelon in order to reduce the problem to a single-echelon location-routing problem. Then clustering operations are performed to assign customers to a combination platform-satellites ("super node"). Finally routing problems are solved for each super-node and the assigned clusters. The second, instead performs the same clustering operations in inverse order.

A fifth approach could be defined decomposing the problem in two single-echelon location routing problems to be solved hierarchically or sequentially, starting form the second echelon.

In the following a tabu search heuristic is presented, where the problem is decomposed in two capacitated facility-location problems and two multi-depot vehicle routing problems. This heuristic is based on the two-phase iterative approach, proposed by Tuzun and Burke [99], and on the nested approach of Nagy and Salhy [79], hence it can be defined as an "iterative-nested approach".

5.2 Introduction to TS

The method of search with tabus, or simply *Tabu Search* (TS), was formalized in 1986 by F. Glover [45], [46], [47], [48]. Contrary to other metaheuristics, the tabu search method is able to use memory and learn lesson from the past.

The guiding principle of the tabu method is simple: the tabu method works with only one current configuration (at the beginning, any solution), which is updated during the successive iterations (DrÉ0 et al. [38]). In each iteration, the mechanism of passage from a configuration S to the next one, S', comprises two stages:

1. Build the set of the neighbors of S, i.e. the set of accessible configurations in only one elementary movement of S. Let $N(S)$ be the set (or the subset) of these neighbors;

2. Evaluate the objective function z of the problem for each configuration belonging to $N(S)$. The configuration S', which succeeds S in the series of the solutions, is the configuration of $N(S)$ in which z takes the minimal value (if it is a minimization problem). Let us note that this configuration S' is adopted even if it is worse than

S, i.e. if $z(S') > z(S)$: due to this characteristic, the tabu method facilitates to avoid the trapping of z in the local minima.

This procedure could be inoperative, because there is the risk to return to a configuration already retained in the previous iterations, so generating cycling phenomenon. In order to avoid this inconvenient it is necessary to have memory of the solutions visited in the last iterations in order to avoid re-visiting them. To this aim at each iteration a list of prohibited movements is updated. This list is referred as *Tabu list* and it contains m movements ($S' \rightarrow S$), which are the opposite of the last m movements ($S \rightarrow S'$). In this way the structure of the neighborhood of a solution depend on the current iteration, i.e. at each iteration k the neighborhood $N(S,k) \subseteq N(S)$, because several solutions, referred as *tabu*, are removed from the set $N(S)$. Tabu moves could be accepted just in case they provide an improvement of the objective function value z (*aspiration criterion*.

The more immediate choice is the memorization of all the visited solutions during the research process, but this idea is not profitable, since it would require too much memory and the checking operation would not be rapid. For this reason a limited set of information is required, referred as *attributes of a move*. These attributes are memorized in one or more *tabu list*. The list represents the *short time memory* of the algorithm, since they have a limited dimension TL, memorizing information only on the last iterations of the research process. The choice of the TL values, referred as *tabu tenure*, can be of two kinds: static and dynamic. The static rules fix the value of the TL in function of the dimension of the problem (suggested values are $\sqrt{n}$, where n is the dimension of the problem). The dynamic rules, instead, randomly choose the tabu tenure values in a range $[TL_{min}, TL_{max}]$. Random tabu tenure generally produce better performances than static rules.

Two additional mechanism, named intensification and diversification, are often implemented to also equip the algorithm with a long time memory. The intensification consists in looking forward into regions of the solution space, identified as particularly promising ones. Diversification is, on the contrary, the periodic reorientation of the search process towards solutions seldom visited. These processes do not exploit the temporal proximity of solutions, but the frequency of their occurrence over a long period.

Hence Tabu search is a metaheuristic procedure which improves the efficiency of a classical local (ascent) search heuristics (LSH or ASH) memorizing information about the research process in order to avoid loops in local optimum. On the other side it does not guarantee any convergence and for this reason a stopping criterion is required.

To conclude the main steps of a *Tabu search* heuristic can be summarized as follows:

- Step 1-Initialization: a starting solution S and its objective function value, or an estimation, are determined.
- Step 2-Neighborhood definition and selection: at each generic iteration k a neighborhood $N(S, k)$ of the current solution S is defined and the solution $S' \in N(S, k)$, associated to the best objective function value, is selected and substituted to the current solution S.
- Step 3-Stopping criterion: If a given stopping criterion is satisfied the algorithm terminates, otherwise set $k = k + 1$ and returns to *Step 2*. The final solution is the best solution found during all the research process.

A general scheme of a TS algorithm is shown in figure 5.1.

5.3 A tabu search heuristic for 2E-LRP

The hardness of location and routing problems is treated in several papers, among which we cite Karp [60], Cornuejols et al. [68] and Lenstra and Rinnooy Kan [68] and it directed a great number of researchers to solve this problem by heuristic methods. For those who are new to the wider research field of location, an extensive list of introductory textbooks and survey papers is given in EWGLA [41]. For vehicle routing, we can recommend Christofides et al. [28], Laporte et al. [63] and Toth and Vigo [96], [97]. To the best of our knowledge, location-routing problems have been approached with exact methods just in the works of Laporte et al. at the beginning of the '80s (see *Section 3.3*). Two-echelon location-routing problem has never been approached until now either by exact nor heuristic methods. The models previously presented are really hard to solve with the usage of a commercial solver. For this reason in the following a tabu search *(TS)* based approach for the two-echelon location-routing problem is proposed. The heuristic sequentially

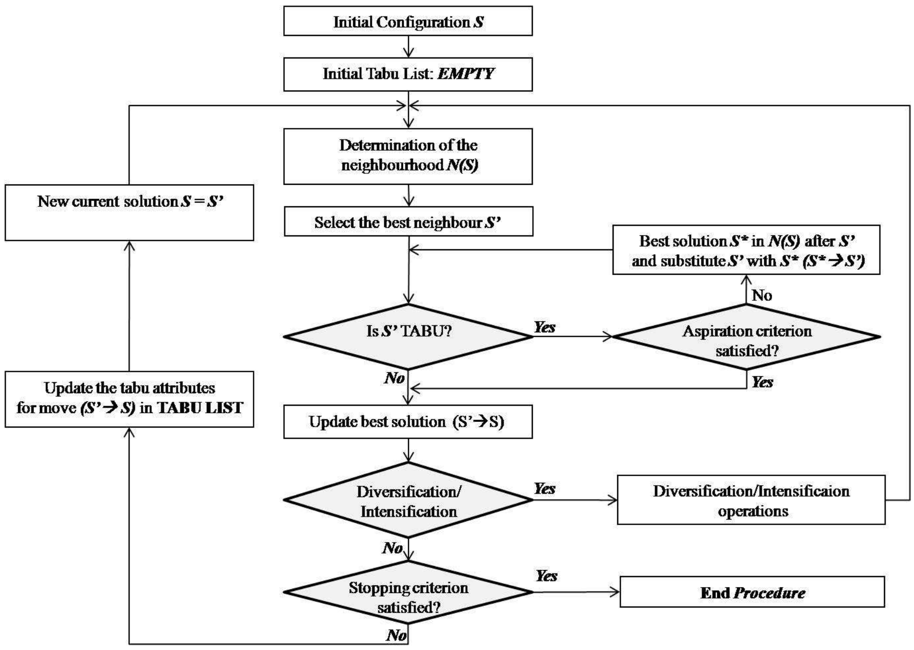


Figure 5.1: Tabu Search scheme.

solves the sub-problems related to each echelon, the two location problems and the two multi-depot vehicle-routing problems, integrating and coordinating the decisions at the different decisional levels.

The TS heuristic proposed for the two-echelon location-routing problem is based on the decomposition of the problem in its two main components, i.e. two location-routing problems. Each component, in turn, is decomposed in the two composing sub-problems, i.e. the capacitated facility location-allocation problem and the multi-depot vehicle routing problem. The heuristic is based on a bottom-up approach, i.e. the first echelon solution is built and optimized on the solution of the second echelon.

The heuristic is structured in two phase for each echelon and integrate the different decisional levels in a computationally efficient manner. Indeed it starts with an initial feasible solution and try to improve it performing the following phases:

1. Location phase: a tabu search is performed on the location variables in order to determine a good configuration of facilities to be

used in the distribution system. The passage from a configuration to another is obtained through the usage of add and swap moves. The two moves are performed sequentially, first swap moves and then add moves. The swap moves keep the number of facilities unchanged but locations change. Swap moves are performed until a maximum number is reached. Then an add move is performed, until a stopping criterion is satisfied.

2. Routing phase: for each location solution determined during the location-phase, a tabu search is performed on the routing variables. The initial routes are built with Clarke and Wright algorithm and then improved by local searches. Finally a tabu search based on insert and swap moves is performed.

The two phases are coordinated and integrated. In fact each time an iteration is performed on the location phase, the routing phase starts in order to update the routing configuration according to the new location solution. For this reason the proposed tabu search can be defined as an iterative-nested approach.

In the following each basic step of the heuristic will be described without keeping into account its relationship with the other ones. Anyway, as already introduced, the key issue of the proposed Tabu Search is the mechanism to combine the solutions of each single sub-problem. For sake of clarity this mechanism will be described after providing all the basic elements, following the scheme proposed in previous section.

5.4 First feasible solution and evaluation

The TS heuristic starts with the construction of a first feasible solution of the two-echelon capacitated facility location-allocation problem (2E-FLP). It is given by two set of selected locations (one for each echelon), and by a feasible assignment satellites-platforms and customers-satellites. Each node is served by a dedicated vehicle, and consequently on a dedicated route. This solution, even if not a good solution, is anyway a feasible solution for the *2E-LRP*.

The *2E-FLP* is a NP-hard problem and it has been widely treated in literature with both exact and heuristic methods. A fast heuristic for the determination of a first feasible solution to this problem is used.

The heuristic is aimed at defining a first feasible solution which uses the minimum number of facilities on both echelons. It starts with the definition of a feasible solution for the location and allocation problem on the second echelon and then it repeats the same operations on the first echelon.

A sorting of the satellites in function of their capacity is performed in order to obtain a list where: $K_1 \geq K_2.... \geq K_S$. The number S^* of satellites to open, in order to fully satisfy the demand of the customers, has to satisfy the following condition:

$$S^* : \alpha \sum_{i=1,..,S^*} K_i = \alpha \left(K_1 + K_2 + ... + K_{S^*} \right) \geq \sum_{z=1,..,Z} D_z \tag{5.1}$$

This formula impose to open the minimum number S^* of satellites such that their total capacity decreased of given percentage α exceeds the total customer demand D_z. This condition, together with the assumption that demand values are much smaller than facility capacities, should guarantee that a feasible assignment of the customers to the satellites could be determined. Generally α varies in the range 90% to 95%. Anyway, in case of unfeasible assignment, a mechanism to increase the number of satellites is foreseen.

Moreover to increase the probability of having a first feasible assignment customers-satellites, which satisfies the capacity constraints for the facilities, customers are sorted in decreasing order of their demand. In this way the customers with the highest demand values are assigned first. The following criteria have been defined to choose the satellite to which a customer has to be assigned:

1. *random*: each customer is assigned randomly to one of the open satellites;

2. *min-distance*: each customer is assigned to the nearest open satellites with residual capacity;

3. *residual capacity*: each customer is assigned to the satellite with the higher residual capacity value.

Therefore to summarize, the minimum number of satellites to open is determined in function of their capacity, whereas the customers are assigned with a criterion based on their demand value.

This procedure can be applied to find a first feasible solution also for the first echelon. In this case, the facilities to open are the platforms and the customers are represented by the satellites. The demand of each satellites is given by the sum of the demands of all the customers assigned to an open satellite, referred as D_s. Therefore, once performed the sorting of the platforms in function of their capacity, the number P^{ast} and the location of the primary facilities has to satisfy the following condition:

$$P^* : \alpha \sum_{i=1,..,P^*} K_i = \alpha\left(K_1 + K_2 + ... + K_P^*\right) \geq \sum_{s \in S} D_s = \sum_{z \in Z} d_z \quad (5.2)$$

The application of this simple heuristic to both echelons returns a solution to the *2E-FLP* where each customer and open satellite are served on dedicated routes. The structure of a first feasible solution is reported in figure 5.2:

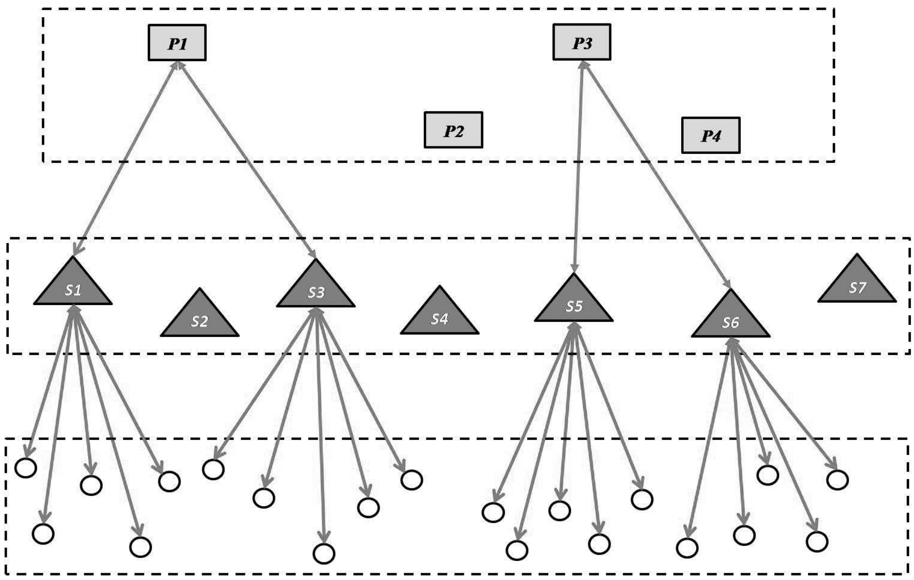


Figure 5.2: First feasible solution.

The idea of opening the minimum number of facilities on both levels is functional to the tabu search strategy and moves that will be performed during the location phase and will be discussed in the following.

It is important to remark that the solution so generated is a first feasible solution for the 2E-LRP, since as already said, capacity constraints for the facilities and single sourcing requirements are satisfied, even if with a great number of vehicles on both echelons, which provide an increase of the total costs of the system. Vehicle costs will be explicitly considered in the evaluation of the goodness of a solution and operations able to reduce this number will be performed in routing phase.

5.4.1 Estimated and actual cost of a solution

In this section the criteria used during the TS for the evaluation of the goodness of a 2E-LRP solution are shown. These criteria will be used in different steps of the procedure.

Two ways to evaluate a solution, referred as *"estimated cost"* and *"actual cost"*, are considered. Both are given by the sum of two components, the location and the routing components (which includes the vehicle costs). Being G^* and V^* respectively the set of first and second echelon used vehicles, P^* and S^{ast} respectively the set of open platforms and satellites, and using the notation introduced for the three-index formulation, the following cost components can be defined:

- *First-echelon location cost*: it is the sum of the location costs of the open platforms:

$$CL_1(P^*) = \sum_{p \in P^*} H_p \; y_p \tag{5.3}$$

- *Second-echelon location cost*: it is the sum of the location costs of the open satellites:

$$CL_2(S^*) = \sum_{s \in S^*} H_s \; y_s \tag{5.4}$$

- *First-echelon estimated routing cost*: it is computed as two times the sum of direct distances between open platforms and assigned satellites. This cost component includes also the cost for the usage of a vehicle of type g:

$$\overline{CR}_1 = 2 \left(\sum_{p \in P^*, s \in S^*} c_{ps} \; r^g_{ps} \sum_{g \in G^*} TCG \; t_g \right) \tag{5.5}$$

- *Second-echelon estimated routing cost*: it is computed as two times the sum of direct distances between the open satellites and the assigned customers. This cost component considers also the cost for the usage of a vehicle of type v:

$$\overline{CR}_2 = 2\left(\sum_{s\in S^*, z\in Z} c_{sz}\ x^v_{sz} \sum_{v\in V^*} TCV\ t_v\right) \tag{5.6}$$

- *First-echelon actual routing costs*: it is obtained each time a routing move is performed to improve the initial solution on the first echelon. This cost component considers also the costs for the usage of a vehicle of type g:

$$CR_1 = \left(\sum_{i,j\in P^*\cup S^*} c_{ij}\ x_{ij} + \sum_{g\in G^*} TCG\ t_g\right) \tag{5.7}$$

- *Second-echelon actual routing costs*: it is obtained each time a routing move is performed to improve the initial solution on the second echelon. This cost component considers also the costs for the usage of a vehicle of type v:

$$CR_2 = \left(\sum_{i,j\in S^*\cup Z} c_{ij}\ x_{ij} + \sum_{v\in V^*} TCV\ t_v\right) \tag{5.8}$$

Therefore the total cost for a generic solution Ψ, referred as $z(\Psi)$, can be expressed in two ways:

$$Estimated\ cost = \overline{z(\Psi)} = CL_1(P^*) + CL_2(S^*) + \overline{CR}_1 + \overline{CR}_2 \tag{5.9}$$

$$Actual\ cost = z(\Psi) = CL_1(P^*) + CL_2(S^*) + CR_1 + CR_2 \tag{5.10}$$

The estimated cost if computed for the first feasible solution obtained of *2E-FLP* and each time a move is performed in the location phase, whereas the actual cost is computed each time a move is performed in the routing phase for one of the two echelons.

5.5 Solution neighborhood definition

The key element of a Tabu Search heuristic is the definition of the neighborhood of a solution and of the tabu rules. In fact a small neighborhood does not allow a good exploration of the solution space, but a large neighborhood could be not effective. In the following the moves to generate the neighborhoods of a solution will be presented and for sake of clarity location and routing moves and the related tabu search parameters will be presented separately.

5.6 Location moves

The location component of the heuristic has to define the number and the location of the facilities to open on each echelon. As already explained, the first feasible solution opens the minimum number of satellites and platforms, since facilities with the highest capacity are chosen, without taking into account the related costs. To explore the solution set, two elementary moves are performed: *swap* and *add* moves.

The two moves are applied sequentially and iteratively on each configuration, but not at the same time. More precisely, for a given number of open facilities, we try to find the best solution changing the combination of open facilities and then, when no improvements are found and a stopping criterion is met, we try to increase the total number of facilities.

We do not consider *"drop"* moves because our strategy explores at the best the solution space for a given number of opens facilities, starting by the minimum number, and therefore reduction mechanism would be not so meaningful.

5.6.1 Swap moves

With this move the status of two facilities (satellites or platforms) is exchanged, i.e. a facility, previously open, is closed, and a facility, previously closed, is opened. Therefore with this operation the number of the open facilities is kept constant. In the application of this move the facility to be opened and the one to be closed have to be opportunely chosen.

Swap moves will be first explained for the satellites and then the mechanism will be extended to the platforms. The key element of these moves is the selection of the facility to be closed and the one to be opened.

The choice of the satellite to be closed in of S^* is based on one of the following criteria:

1. *Rand-sel-out*: random selection of a node belonging to the solution set S^*g to the solution set S^*
2. *Max-cost*: the node of S^* associated with the highest location and routing cost (weighted with the number of served customers):
$$\frac{\sum_{j=1}^{Z} c_{sj}\ x_{sj}}{Z_s} + \ H_s\ y_s \tag{5.11}$$
In this expression Z_s is the number of customers served by a satellite s.
3. *Max-route*: the node of S^* associated with the highest cost for a single route.
4. *Max-loc*: The node of S^* associated with the highest location cost.

Concerning instead the choice of the candidate set of satellites to be opened, it has to be done considering just the nodes satisfying the previous introduced relation for the demand of the customers. This means that being S^* the node solution set for satellites, i^* the node selected to be removed from S^* and j^* the node to be inserted, then:

$$\alpha \sum_{i \in S^* \setminus \{i^*\} \cup \{j^*\}} K_i = \alpha \left(K_1 + K_2 + .. - K_i^* + ... + K_j^* + K_S^*\right) \geq \sum_{z=1,..,Z} d_z \tag{5.12}$$

Once determined the set of candidate satellites, two criteria for the determination of the choice of the entering satellite among the candidate ones are used:

1. *Rand-sel-in*: random selection of a node in the candidate set.
2. *Min-cost*: introduction in the solution of the node associated with the minimum estimated total cost
$$CL1(P^*) + CL2(S') + \overline{CR1} + \overline{CR2} \tag{5.13}$$
where S' indicates the new solution for the satellites location.

In this way, when the number of open satellites is not very high, solutions characterized by facilities with similar capacity values are considered, so reducing the number of possible exchange moves to explore. This means that at the beginning, when the number of open facilities is small, we will explore solutions where the facilities with highest capacity values are open. Whereas increasing the number of open satellites, we will explore solution characterized by capacity values which could be very different from a facility to another.

Once a swap move has been performed, the two facilities are declared *tabu* for a number of iteration which depends on the number of open facilities. More precisely the tabu tenure, $tabu - swap - loc - s$, for satellite swap location moves, will be:

$$tabu - swap - loc - s = \alpha \, |S^*|$$

where α is a random value in the range $[\alpha_{min} \div \alpha_{max}]$.

If a move is tabu it is not performed. A simple aspiration criterion is used. If an improvement of the total actual cost for the system is obtained, then the solution is updated also if it is tabu.

Swap moves are performed until a maximum number of iterations without improvement is reached. This value is fixed and it will be referred as $max - swap - loc - s$.

Concerning the swap moves for the platforms, they are applied for each satellite configuration, i.e. for each set of open facilities characterized by their demand, given by the sum of the demand of the customers assigned to it.

The criteria for the choice of the node to be swapped and the entering one are basically the same presented for the satellites, so as the aspiration criteria, definition of the tabu tenure values and maximum number of moves without improvements, but they depend on the open platform set. Therefore the following relations are defined:

$$tabu - swap - loc - p = \alpha \, |P^*|$$

where α is a random value in the range $[\alpha_{min} \div \ alpha_{max}]$.

Also for the platforms, swap moves are performed until the maximum number of iterations without improvement $max - swap - loc - p$ is reached.

5.6.2 Add move

With this move the number of open facilities is increased. After a prefixed number of swap moves without improvement, we add a facility. The increased number of open facilities could provide a reduction of the transportation costs, which overcomes the additional location cost and moreover allows to open smaller facilities, characterized by lower location costs. The criterion that we use to decide which is the entering node, is the same used for the swap moves. It has to be a facility that, added to the current best solution, is associated with the minimum estimated cost.

When we perform an add move the facility introduced in the solution set is declared tabu for a predefined number of add moves. The tabu tenure values depend on the overall number of available locations for satellites and platforms ($|P|$ and $|S|$):

$$tabu - add - loc - s = \alpha\,|S|$$

$$tabu - add - loc - p = \alpha\,|P|$$

where α is a random value in the range $[\alpha_{min} \div \alpha_{max}]$.

We use random tabu tenure values because randomness generally guarantee better solutions [38]. Concerning instead the stopping criterion for this move, it is a value fixed at the beginning of the TS and referred as $max - add - loc - s$ and $max - add - loc - p$, respectively for satellites and platforms.

5.7 Routing moves

At this step, starting from the first feasible solution, we perform several operations to improve the routing component. Starting from the first feasible assignment, the definition and the optimization of the routes is based on three phases:

1. *Phase-1*: definition of multi-stop routes and improvements of a single route assigned to a *single* facility:
   - Clarke and Wright algorithm [32];
   - 2-opt and 3-opt algorithms [69], [71].

2. *Phase-2*: optimization of *multiple* routes assigned to a *single* facility:
   - insert moves for a single facility;
   - swap moves for a single facility.
3. *Phase-3*: optimization *multiple* routes assigned to *multiple* facilities:
   - insert moves for multiple facilities;
   - swap moves for multiple facilities;

Therefore we sequentially try to improve the routing costs acting locally on each route and then expanding the research process. These phases have different effects on the global solution of the problem. Indeed the first and the second phase optimize the routing cost component but do not affect the assignment of the customer to the satellites, i.e. they do not affect the value of the demand D_s assigned to each satellite. Consequently the routing costs on the first echelon remains unchanged. On the other side, third phase can provide significant changes of the demand D_s assigned to a satellites involved in the insertion or in the swap moves. Therefore they can affect also the assignment problem at the first echelon and consequently the routing solution. The used approach to solve these changes in the global solution will be explained in the following. Here a brief discussion about the used moves is provided.

5.7.1 Multi-stop routes definition and improvement

In this phase we try to pass from the first feasible solution, where we have just the assignment of the customers to the satellites and of the satellites to the platforms, to a solution where we have multi-stop routes to serve the customers and the satellites. Two classical algorithms for the VRP are used respectively to generate and optmise the routes: *Clarke and Wright*, *2-opt* and *3-opt* algorithms:

- *Clarke and Wright, C&W, [32]*: it is a saving based algorithm. It is applied when the number of vehicles is not known. This algorithm is based on the definition of *savings*. When two routes $(0, ..., i, 0)$ and $(0, j, ..., 0)$ can feasibly be merged into a single route $(0, ..., i, j, ..., 0)$, a distance saving $s_{ij} = c_{i0} + c_{0j} - cij$, where c_{ij}

indicate the euclidean distance between the customer j and the customer i, is generated. A parallel and a sequential version are available. In the proposed TS the parallel version has been implemented. The main steps of this algorithm can be summarized as follows:

1. *Step 1: Saving Computation.* Compute the savings for $i, j \in N$. Create $|N|$ vehicle routes $(0, i, 0)$ for $i = 1...|N|$ and order the savings in a non increasing order.
2. *Step 2: Best Feasible Merge.* Starting from the top of the savings list, execute the following:
   (a) Given a saving s_{ij}, determine whether there exist two routes one containing arc $(0, j)$ and the other containing arc $(i, 0)$, that can feasibly be merged
   (b) Combine these two routes by deleting arc $(0, j)$ and arc $(i, 0)$ and introducing arc (i, j) .

- *2-opt and 3-opt algorithm*: k opt algorithms is probably the most popular improvement heuristic for the VRP, introduced by Lin [69] and extended by Lin and Kerninghan [71]. A *k-change* algorithm consists of deleting k edges and replacing them by k other edges to form a new route. The heuristic procedure begins with any feasible route. From this route, all possible *k-changes are examined.* If a route is found that has a lower cost than the current solution, it becomes the new solution. The process is repeated until no further *k-change* results in a better solution. When the algorithm stops, we have a local optimal solution. Of course there is no guarantee that the resulting solution is globally optimal. Higher values of k provide higher number of exchanges, i.e. for example *3-change* heuristic will find a better solution than a *2-change* heuristic will. However the computational cost of enumerating all *3-changes* is larger than the cost of enumerating all *2-changes.* One must balance the value of finding better solution against the increased computational effort. In the proposed heuristic, to limit this effort, a special case of *3-opt* moves is considered. In fact we consider the case where two of the removed arcs are adjacent and the third disconnected. In this way, there is just one possible reconnection of the subcycles.

5.7.2 Intra-routes improvements for a single facility

In order to improve the routing of a single facility, two kind of moves have been performed. We perform insert and swap moves of the nodes assigned to a single facility.

The main issue for these moves is the choice of the node (nodes) to use in the definition of the neighborhood, i.e. respectively the node to insert in another route and the node to be swapped. Three selection criteria are used to restrict the sizes of neighborhoods:

1. *One-select*: select one node and evaluate the related neighborhood;
2. *Path-select*: select randomly a path and evaluate for all the nodes the related neighborhood;
3. *Perc-sel*: select a percentage of all the nodes to be served on an echelon and evaluate the related neighborhood.

In *insert moves*, the selected node (nodes) is inserted in all the routes assigned to the same facility. In *swap moves* each node (nodes) is exchanged with the whole set of nodes assigned to the same facility.

For sake of clarity, the moves will be presented referring to second echelon and then the used criteria are extended to the first echelon. In the following the notation $CR2^*$ for the best found solution on the second echelon and $CR2'$ for the solution determined after a move will be used.

- Insert move: A customer is deleted from one route and it is assigned to another route belonging to the same satellite. The neighborhood is given by all the customers assigned to the satellite.

The insertion is feasible just if it satisfies vehicle capacity constraints. Unfeasible moves are not allowed. If a neighbor solution provides an improvement, i.e. $CR2' \leq CR2^*$ then the move is performed and the added node is declared tabu. If no neighbor solution provides an improvement, i.e. $CR2' \geq CR2^*$ then we implement the best non-tabu deteriorating move and the added node is declared tabu. A move is implemented, even if tabu, if it provides an improvement of the solution (*aspiration criterion*).

Tabu tenure value, $tabu - r - ins - single - s$ is variable and it depends on the number of customers assigned to a satellite. Being Z_s the total number of customers assigned to satellite S, then this value is computed as:

$$tabu - r - ins - single - s = \lceil \alpha \ Z_s \rceil \quad (5.14)$$

where α is a random value chosen in the range $[\alpha_{min} \div \alpha_{max}]$.

This move is performed until the fixed maximum number, $max-r-ins-single$ of not-improving moves is reached.

The extension for the platforms in the following way:

$$tabu - r - ins - single - p = \lceil \alpha \ S_p \rceil \quad (5.15)$$

where α is a random value chosen in the range $[\alpha_{min} \div \alpha_{max}]$ and S_p is the total number of satellites assigned to the platform p.

Therefore the same relations are used for both echelons. The difference in in the fact that the sizes of the involved sets, Z, S and P are different and this affects the tabu tenure values.

It is important to underline that in the evaluation of the routing costs deriving from an insert move, we consider also the possibility that if we have a route with a single customer, then its insertion in another route provides a saving equal to the cost of the vehicle. Therefore in this way the minimization of the number of vehicles is obtained.

- Swap moves: We choose two customers belonging to two routes assigned to the same satellite and we try to exchange their position on their routes. If the exchange satisfy the capacity constraints for the two vehicles involved in the exchange move, then the move is allowed. As it happened for the insertion moves, if the move is not-tabu and it provides a saving on the routing cost, it is immediately performed and the value of $CR2^*$ is updated. Otherwise we perform the best non-tabu deteriorating move, but without updating the $CR2^*$ value. Also in this case a move is performed if it is tabu but it provides an improvement on the best known solution of the routing component (*aspiration criterion*). The nodes used in the exchange moves are both declared tabu for a number of iteration that is variable and is determined with the previous introduced relation for the insert moves:

$$tabu - r - swap - single - s = \lceil \alpha \ Z_s \rceil \quad (5.16)$$

$$tabu - r - swap - single - p = \lceil \alpha \ S_p \rceil \quad (5.17)$$

These moves are performed until the fixed maximum number, $max-r-swap-single$ of not-improving moves is reached.

5.7.3 Intra-routes improvements for multiple facilities

With these moves we try to improve the routing cost operating on routes assigned to two different facilities. The moves are the ones previously presented, insert and swap moves, but in this case, a move to be feasible has to satisfy two criteria: at first it has to satisfy the capacity constraints related to the facilities involved in the move and then it has to satisfy the capacity constraints of the vehicles.

In the definition of the neighborhood the same selection criteria presented for the previous moves are used. The difference is in the choice of the routes to use for the insertion and the choice of the nodes to use in the swap. These issues will be discussed in detail in the following two sub-sections. As previously done, these moves will be presented referring to the second echelon and then extended to the first one.

- Insert move: A customer is randomly chosen and it is deleted from its route. Then we try to insert it in another route belonging to another open satellite. The move can be performed if and only if the satellite and the vehicle to which the customer will be assigned have still enough residual capacity. Therefore no dedicated route can be used to serve the inserted customer. If a move is not-tabu and it provides an improvement, it is immediately performed. Otherwise if it provides no improvement, the best deteriorating non-tabu one is performed.

In this case it is important to restrict the neighborhood of a solution. In fact considering all the possible insertions, computation time would exponentially increase in large instances. Therefore we limit our search trying to insert the selected customer just in routes belonging to the "closest" open satellite, i.e. a percentage of all the open satellites. This value, $near - ins - s$ is defined as a percentage of the open satellites at that iteration: $near - ins - s = \lceil \beta \ S^* \rceil$, where $\beta \in [0 \div 1]$.

Tabu tenure value for this move is given by a relation similar to the one previously introduced for the insertion move for a single facility, but in this case the value depend on the total number of customers Z:

$$tabu - r - ins - multi - s = \lceil \alpha \ Z \rceil \tag{5.18}$$

where α is a random value chosen in the range $[\alpha_{min} \div \alpha_{max}]$.

The same relations can be extended to the first echelon, for which we have $near - ins - p = \lceil \beta \ P^* \rceil$, $,\beta \in [0, 1]$ and the following tabu tenure value:

$$tabu - r - ins - multi - p = \lceil \alpha \ S \rceil \tag{5.19}$$

where α is a random value chosen in the range $[\alpha_{min} \div \alpha_{max}]$ and S is the total number of satellite locations.

This move is performed until the fixed max number, $max-r-ins-multi$ of not-improving moves is reached. Once performed an insertion move, a local search is used to re-optimise locally the routes of the two involved satellites, i.e. 2-opt, 3-opt and insert and swap moves for a single facility.

- Swap move: We randomly choose two customers belonging to routes assigned to two different satellites. We check if the exchange of position between the two customers is feasible, i.e. it satisfies the capacity constraints for the two satellites involved in the move and for the two trucks involved in the move. If the move is feasible, it provides an improvement and it is not-tabu, then the move is performed and the objective function value is updated, otherwise the best deteriorating non-tabu move is performed.

Also in this case it is important to efficiently define the neighborhood of a solution. Therefore we restrict our search trying to swap two customers just if they are "close". To obtain this, we try to swap a customer just with its nearest nodes. The number of nearest customers to consider, referred as $near-swap-s$ is defined as a percentage of the total customers:$near-swap-s = \lceil \beta\ Z \rceil$, where $\beta \in [0 \div 1]$.

Tabu tenure value for this move is given by a relation equal to the one previously introduced for the insertion move:

$$tabu-r-swap-multi-s = \lceil \alpha\ Z \rceil \tag{5.20}$$

where α is a random value chosen in the range $[\alpha_{min} \div \alpha_{max}]$.

The same relations can be extended to the first echelon, for which we have $near-swap-p = \lceil \beta\ S^* \rceil$, ,$\beta \in [0 \div 1]$ and the following tabu tenure value:

$$tabu-r-swap-multi-p = \lceil \alpha\ S \rceil \tag{5.21}$$

where α is a random value chosen in the range $[\alpha_{min} \div \alpha_{max}]$ and S is the total number of satellite locations.

This move is performed until the fixed max number, $max-r-swap-multi$ of not-improving moves is reached. Also in this case, once performed a swap move, a local search is used to re-optimise locally the routes of the two involved satellites, i.e. 2-opt, 3-opt and insert and swap moves for a single facility.

5.8 Combining sub-problems

After the determination of the first feasible solution, as already said we have an assignment of the customers to the satellites and of the satellites to the platforms. Applying the previous moves, location and routing, on each echelon, we locally optimise the four sub-components of the system. At this point the key elements are the mechanism to combine the location and routing solutions on a single echelon, and the mechanism to combine the solutions of the two echelons in order to obtain a solution that is globally good.

5.8.1 Combining sub-problems of a single echelon

In order to find a good solution for the capacitated location-routing problem related to a single echelon, the idea proposed in Tuzun and Burke [99] is adopted (figure 5.3). In the location phase of the algorithm, a TS is performed on the location variables, starting from the configuration with the minimum number of open facilities. For each of the location configuration, another TS is run on the routing variables in order to obtain a good routing for the given configuration. Therefore each time a move is performed on the location phase, the routing phase is started in order to update the routing according to the new configuration.

5.8.2 Combining sub-problems of the two echelons

We decomposed the 2E-LRP in two capacitated location-routing problems, one for each echelon. The two problems are solved separately but not in a pure sequential way. In fact, in sequential approaches, the two problems are solved just once starting from the bottom level. On the contrary our approach foresees the resolution of the four problems several times, in order to explore different location solution combinations of the first and the second echelon. In practice each time a change of the demand assigned to a set of open satellites occurs, i.e. each time a routing move for multiple satellites is performed, then the location-routing problem of the first echelon should be re-solved in order to find the best location and routing solution to serve the new demand. Three criteria have been defined to control the return on the first echelon:

1. *Always*: return to *Step2* each time an intra satellite move is performed on the second echelon.

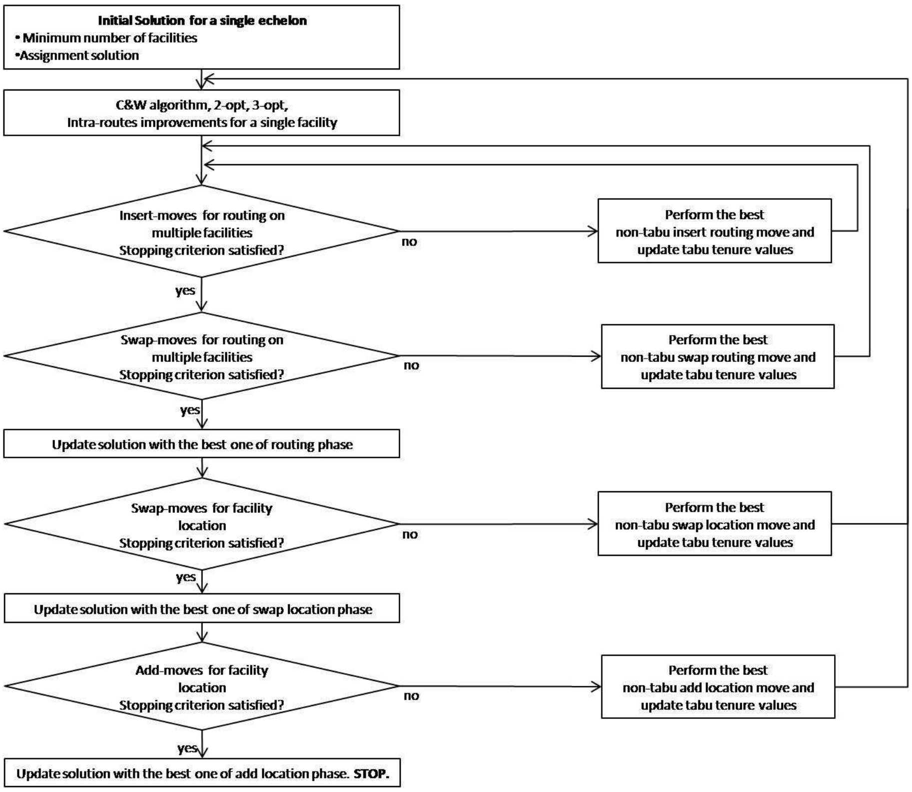


Figure 5.3: Combining sub-problems on a single-echelon.

2. *Imp-CR2*: return to *Step2* just if a better solution for the second echelon routing problem has been determined, otherwise continues to explore the second echelon.

3. *Violated-cap*: return to *Step2* if an improvement of the routing for the second echelon is obtained and if the capacity constraints for the first echelon (platform or vehicles) of the previous solution are violated by this new set of demands.

Therefore the main steps of the Tabu Search can be summarized as follows:

- *Step 0*: solve the two-level capacitated facility location problem and determine the minimum number of open facilities on each echelon.

- *Step 1*: determine the demand assigned to the open satellites and platforms. Compute the estimated cost of the solution $z(\Psi)$ and go to *Step 2*.

- *Step 2*: define and optimize multi-stop routes on the first echelon with C&W algorithm, 2-opt and 3-opt algorithms.

- *Step 3*: perform insert moves for a single open platform. If the maximum number of insert moves without improvement ($max - r - ins - single$), is reached, then update solution with the best determined one and go to *Step 4*. Otherwise repeat *Step 3*.

- *Step 4*: perform swap moves for a single open platform. If the maximum number of swap moves without improvement ($max - r - swap - single$) is reached, then update solution with the best determined one and go to *Step 5*. Otherwise repeat *Step 4*.

- *Step 5*: perform insert moves for multiple platforms. If the maximum number of insert moves without improvement ($max - r - swap - multi$) is reached, then update the cost of the solution and go to *Step 6*. Otherwise repeat *Step 5*.

- *Step 6*: perform swap moves for multiple platforms. If the maximum number of swap moves without improvement ($max - r - swap - multi$) is reached, then update the cost of the solution and go to *Step 7*. Otherwise repeat *Step 6*.

- *Step 7*: perform a swap location move for the first echelon. If the maximum number of swap moves without improvement ($max - add - loc - p$) is reached, then update solution cost and go to *Step 8*. Otherwise return to *Step 1*.

- *Step 8*: perform an add move for the second echelon. If the maximum number of add moves without improvement ($max - swap - loc - p$) is reached , then update solution cost and go to *Step 9*. Otherwise return to *Step 1*.

- *Step 9*: define and optimize multi-stop routes on the second echelon with C&W algorithm, 2-opt and 3-opt algorithms and go to *Step 10*.

- *Step 10*: perform insert moves for a single open satellite. If the maximum number of insert moves without improvement ($max - r - ins - single$), is reached, then update solution with the best determined one and go to *Step 11*. Otherwise repeat *Step 10*.

- *Step 11*: perform swap moves for a single open satellite. If the maximum number of swap moves without improvement ($max - r - swap - single$) is reached, then update solution with the best determined one and go to *Step 12*. Otherwise repeat *Step11*.

- *Step 12*: perform insert moves for multiple satellites. If one of the criteria, *Always*, *Imp-CR2* or *Violated-cap* is satisfied, return to *Step 1*, otherwis if the maximum number of insert moves without improvement ($max - r - swap - multi$) is reached, then update the cost of the solution and go to *Step 13*. Otherwise repeat *Step 12*.

- *Step 13*: perform swap moves for multiple satellites. If one of the criteria, *Always*, *Imp-CR2* or *Violated-cap* is satisfied, return to *Step 1*, otherwise if the maximum number of insert moves without improvement ($max - r - swap - multi$) is reached, then update the cost of the solution and go to *Step 14*. Otherwise repeat *Step 13*. item *Step 14*: perform a swap location move for the second echelon. If the maximum number of swap moves without improvement is reached ($max - swap - loc - s$), then update solution cost and go to *Step 15*. Otherwise return to *Step 1*.

- *Step 15*: perform an add move for the second echelon. If the maximum number of add moves without improvement ($max - add - loc - s$) is reached, then update solution cost and STOP. Otherwise return to *Step 1*.

5.9 Diversification criteria

A simple diversification criterion has been considered in order to better explore the solution space of the 2E-LRP. The diversification is applied on the location variables of both echelons during the swap moves of the location phase. Therefore it should be introduced in *Step 2* and in *Step 4* of the functioning mechanism of the TS. The criterion works as follows. When a prefixed number of swap location moves without improvement is reached, $div - val - s$ and $div - val - p$ respectively for satellites and platforms, we force a change in the set of open facilities on both echelons. Therefore we close the facilities which appear more frequently in the explored solutions and we open the less present ones. It is important to note that $div - val - s$ has to be lower than $max - swap - locs$ and $div-val-p$ lower than $max-swap-loc-p$, otherwise the diversification will not be performed.

We also introduced two values, $max - freq - s$ and $max - freq - p$, in order to privilege the facility locations, with an occurrence frequency lower than these fixed values.

Chapter 6

Computational results of TS for 2E-LRP

In this chapter the results of the Tabu Search heuristic on three sets of instances are presented. The three sets differ for the spatial distribution of satellites. Each instance has been solved with four settings of the tabu search parameters. TS heuristic has been developed in C++ and instances were run on an Intel(R) Pentium(R) 4(2.40 GHz, RAM 4.00 GB). Results have been compared with the solutions provided by a commercial solver in terms of quality and computation times.

6.1 TS results and settings

TS heuristics require an important tuning phase for the parameters in order to be effective. The number of parameters of the proposed TS is huge and they are summarized in tables 6.1 and 6.2.In the following we will not report the results obtained with all the experienced parameter settings of the TS, but we will concentrate on four of them which provided good results in terms of quality of solutions and computation times. The four considered TS settings will be respectively referred as *TS1*, *TS2*, *TS3*, *TS4*. They differ for the size of the related neighborhood and tabu tenure values, which are reported in tables 6.3 and 6.4.

TS results for three set of instances of varying dimensions are reported in Tables 6.5, 6.6, 6.7. For each setting the related best solution values (*TS1*, *TS2*, *TS3*, *TS4*) and computation times in seconds (*CPU-1*, *CPU-2*, *CPU-3*, *CPU-4*) are reported.

Rand-sel-out	random selection of the node to be swapped
Max-cost	selection of the node to be swapped associated to max estimated cost
Max-route	selection of the node to be swapped associated to max route cost
Max-loc	selection of the node to be swapped associated to max location cost
Rand-sel-in	random selection of the entering node
Min-cost	selection of the entering node associated to min estimated cost
One-select	neighborhood of a single randomly selected node
Path-select	neighborhood of the nodes of a single randomly selected path
Perc-sel	neighborhood of a percentage of the total number of nodes for each echelon
Always	return on the first echelon every time a routing move is performed on second echelon
Imp-CR2	return on the first echelon just if an improvement of second echelon routing is found
Violated-cap	return on the first echelon just if an improvement of second echelon routing is found and capacity constraints are violated

Table 6.1: Tabu Search criteria.

From tables 6.5, 6.6, 6.7 we can observe that from *setting 1* to *setting 4* results are characterized by increasing computation time and increasing quality of solutions. Results of *setting 2* and *3* are very similar, whereas results of *setting 1* and *4* present opposite characteristics in terms of quality of solutions and computation times. In any case computation times are lower than *3600* seconds and just for instances with more than 150 customers they increase until about *7200* seconds. It is important to observe that the higher computation times are related to test set *I2*. This is probably due to the distribution of the satellites, which, in this case, is uniform in all the area under investigation (for details see the appendix). In the following we will concentrate on *Setting 1* and *Setting 4* to evaluate the goodness of the TS in the worst and the best case. Before comparing TS results with the bounds deriving from the models and decomposition approach, we report in tables 6.8 and 6.9, the results obtained with *setting 1* and *4* with the usage of the diversification criterion (*DTS1*, *DTS4*) on the medium and large instances and the related computation time (*CPU-DTS1*, *CPU-DTS4*). To perform the diversification, the following value have been imposed:

- *Setting 1*: $div-val-s = 2$, $div-val-p = 0$, $max-freq-s = 2$, $max-freq-p = 1$;
- *Setting 4*: $div-val-s = 4$, $div-val-p = 2$, $max-freq-s = 3$, $max-freq-p = 2$;

tabu-swap-loc-s	tabu tenure value for satellite swap moves $\alpha\,\lvert S^*\rvert\,,[\alpha_{min} \div \alpha_{max}]$
max-swap-loc-s	maximum number of iterations without improvement for satellite swap move
tabu-swap-loc-p	tabu tenure value for platform swap moves $\alpha\,\lvert P^*\rvert\,,[\alpha_{min} \div \alpha_{max}]$
max-swap-loc-p	maximum number of iterations without improvement for platform swap move
tabu-add-loc-s	tabu tenure value for satellite add moves $\alpha\,\lvert S*\rvert\,,[\alpha_{min} \div \alpha_{max}]$
max-add-loc-s	maximum number of iterations without improvement for satellite add move
tabu-add-loc-p	tabu tenure value for platform add moves $\alpha\,\lvert P*\rvert\,,[\alpha_{min} \div \alpha_{max}]$
max-add-loc-p	maximum number of iterations without improvement for satellite add move
tabu-r-ins-single-s	tabu tenure value for single satellite routing insertion moves $\lceil\alpha\ Z_s\rceil\,,[\alpha_{min} \div \alpha_{max}]$
tabu-r-ins-single-p	tabu tenure value for single platform routing insertion moves $\lceil\alpha\ S_p\rceil\,,[\alpha_{min} \div \alpha_{max}]$
max-r-ins-single	maximum number of iteration without improvement for single facility routing insert moves
tabu-r-swap-single-s	tabu tenure value for single satellite routing swap moves $\lceil\alpha\ Z_s\rceil\,,[\alpha_{min} \div \alpha_{max}]$
tabu-r-swap-single-p	tabu tenure value for single platform routing swap moves $\lceil\alpha\ S_p\rceil\,,[\alpha_{min} \div \alpha_{max}]$
max-r-swap-single	maximum number of iteration without improvement for single facility routing swap moves
near-ins-s	percentage of all the open satellites for insert routing moves for multiple facilities $\lceil\beta\ S^*\rceil\,,\beta \in [0 \div 1]$
tabu-r-ins-multi-s	tabu tenure value for multiple satellites routing insertion moves $\lceil\alpha\ Z\rceil\,,[\alpha_{min} \div \alpha_{max}]$
near-ins-p	percentage of all the open platforms for insert routing moves for multiple facilities $\lceil\beta\ P^*\rceil\,,\beta \in [0 \div 1]$
tabu-r-ins-multi-p	tabu tenure value for multiple satellites routing insertion moves $\lceil\alpha\ Z\rceil\,,[\alpha_{min} \div \alpha_{max}]$
max-r-ins-multi	maximum number of insertion moves for multiple facilities
near-swap-s	percentage of all the open satellites for insert routing moves for multiple facilities $\lceil\beta\ Z\rceil\,,\beta \in [0 \div 1]$
tabu-r-swap-multi-s	tabu tenure value for multiple satellites routing swap moves $\lceil\alpha\ Z\rceil\,,[\alpha_{min} \div \alpha_{max}]$
near-swap-p	percentage of all the open satellites for swap routing moves for multiple facilities $\lceil\beta\ S\rceil\,,\beta \in [0 \div 1]$
tabu-r-swap-multi-p	tabu tenure value for multiple platforms routing swap moves $\lceil\alpha\ S\rceil\,,[\alpha_{min} \div \alpha_{max}]$
max-r-swap-multi	maximum number of insertion moves for multiple facilities
div-val-s	diversification criterion for satellites
div-val-p	diversification criterion for satellites
max-freq-s	max frequency value in diversification for satellites
max-freq-p	max frequency value in diversification for satellites

Table 6.2: Tabu Search parameters.

TS set 1		TS set 2	
Rand-sel-out	true	*Rand-sel-out*	true
Min-cost	true	*Min-cost*	true
Perc-sel	0.10	*Perc-sel*	0.50
Violated-cap	true	*Violated-cap*	true
tabu-swap-loc-s	[25% ÷ 50%]	*tabu-swap-loc-s*	[30% ÷ 90%]
max-swap-loc-s	4	*max-swap-loc-s*	4
tabu-swap-loc-p	[25% ÷ 50%]	*tabu-swap-loc-p*	[30% ÷ 80%]
max-swap-loc-p	2	*max-swap-loc-p*	4
tabu-add-loc-s	[15% ÷ 30%]	*tabu-add-loc-s*	[20% ÷ 50%]
max-add-loc-s	3	*max-add-loc-s*	3
tabu-add-loc-p	[15% ÷ 30%]	*tabu-add-loc-p*	[10% ÷ 30%]
max-add-loc-p	3	*max-add-loc-p*	3
tabu-r-ins-single-s	[30% ÷ 80%]	*tabu-r-ins-single-s*	[20% ÷ 50%]
tabu-r-ins-single-p	[30% ÷ 80%]	*tabu-r-ins-single-p*	[20% ÷ 50%]
max-r-ins-single	3	*max-r-ins-single*	3
tabu-r-swap-single-s	[30% ÷ 80%]	*tabu-r-swap-single-s*	[30% ÷ 60%]
tabu-r-swap-single-p	[30% ÷ 80%]	*tabu-r-swap-single-p*	[20% ÷ 50%]
max-r-swap-single	3	*max-r-swap-single*	3
near-ins-s	0.10	*near-ins-s*	0.30
tabu-r-ins-multi-s	[10%; 15%]	*tabu-r-ins-multi-s*	[10%; 30%]
near-ins-p	0.10	*near-ins-p*	0.30
tabu-r-ins-multi-p	[10% ÷ 15%]	*tabu-r-ins-multi-p*	[10% ÷ 30%]
max-r-ins-multi	5	*max-r-ins-multi*	5
near-swap-s	0.10	*near-swap-s*	0.15
tabu-r-swap-multi-s	[10% ÷ 15%]	*tabu-r-swap-multi-s*	[10% ÷ 30%]
near-swap-p	0.10	*near-swap-p*	0.25
tabu-r-swap-multi-p	[10% ÷ 15%]	*tabu-r-swap-multi-p*	[10% ÷ 30%]
max-r-swap-multi	3	*max-r-swap-multi*	5

Table 6.3: Tabu Search setttings 1 and 2.

TS set 3		TS set 4	
Rand-sel-out	true	*Rand-sel-out*	true
Min-cost	true	*Min-cost*	true
Perc-sel	0.25	*Perc-sel*	0.50
Violated-cap	true	*Violated-cap*	true
tabu-swap-loc-s	[50% ÷ 75%]	*tabu-swap-loc-s*	[30% ÷ 80%]
max-swap-loc-s	5	*max-swap-loc-s*	7
tabu-swap-loc-p	[30% ÷ 60%]	*tabu-swap-loc-p*	[30% ÷ 50%]
max-swap-loc-p	3	*max-swap-loc-p*	5
tabu-add-loc-s	[30% ÷ 50%]	*tabu-add-loc-s*	[10% ÷ 30%]
max-add-loc-s	3	*max-add-loc-s*	5
tabu-add-loc-p	[10% ÷ 30%]	*tabu-add-loc-p*	[10% ÷ 30%]
max-add-loc-p	3	*max-add-loc-p*	5
tabu-r-ins-single-s	[30% ÷ 100%]	*tabu-r-ins-single-s*	[20% ÷ 50%]
tabu-r-ins-single-p	[30% ÷ 80%]	*tabu-r-ins-single-p*	[20% ÷ 50%]
max-r-ins-single	5	*max-r-ins-single*	5
tabu-r-swap-single-s	[30% ÷ 100%]	*tabu-r-swap-single-s*	[30% ÷ 80%]
tabu-r-swap-single-p	[30% ÷ 80%]	*tabu-r-swap-single-p*	[30% ÷ 80%]
max-r-swap-single	5	*max-r-swap-single*	5
near-ins-s	0.25	*near-ins-s*	0.50
tabu-r-ins-multi-s	[10% ÷ 15%]	*tabu-r-ins-multi-s*	[5% ÷ 25%]
near-ins-p	0.30	*near-ins-p*	0.50
tabu-r-ins-multi-p	[10% ÷ 15%]	*tabu-r-ins-multi-p*	[5% ÷ 25%]
max-r-ins-multi	7	*max-r-ins-multi*	7
near-swap-s	0.25	*near-swap-s*	0.25
tabu-r-swap-multi-s	[10% ÷ 15%]	*tabu-r-swap-multi-s*	[5% ÷ 25%]
near-swap-p	0.25	*near-swap-p*	0.50
tabu-r-swap-multi-p	[10% ÷ 15%]	*tabu-r-swap-multi-p*	[5% ÷ 25%]
max-r-swap-multi	5	*max-r-swap-multi*	7

Table 6.4: Tabu Search settings 3 and 4.

Instance	TS1	CPU-1	TS2	CPU-2	TS3	CPU-3	TS4	CPU-4
I1-238	591.83	0.39	591.83	0.55	591.83	0.42	591.83	0.50
I1-239	902.45	0.59	902.45	0.49	902.45	0.43	878.69	1.08
I1-248	625.96	0.85	625.96	0.94	625.96	0.08	625.96	1.67
I1-2410	862.91	0.85	862.91	1.53	862.91	0.80	862.91	4.07
I1-2415	1121.50	1.92	1115.98	1.68	1116.95	1.31	1105.67	4.15
I1-3510	952.86	1.25	932.67	1.83	952.89	1.34	829.25	5.29
I1-3515	1068.00	2.21	1068.00	3.07	1070.93	2.05	1019.57	6.16
I1-2820	1114.41	5.28	1059.41	23.77	1051.59	26.81	1055.20	48.22
I1-2825	1021.69	4.00	1024.98	14.14	1024.98	31.14	979.85	35.91
I1-21015	754.63	1.53	732.48	7.79	732.48	11.40	732.48	10.81
I1-21020	1008.17	3.40	1003.94	15.27	982.56	50.90	947.65	51.94
I1-21025	1085.67	6.81	1085.67	29.04	1071.63	68.82	1084.26	86.17
I1-3810	604.37	1.24	606.68	3.68	604.37	5.72	604.37	11.26
I1-3815	730.36	1.61	730.36	4.51	730.36	7.10	730.36	11.12
I1-3820	968.59	5.54	947.54	28.77	892.05	63.38	898.08	154.62
I1-3825	943.25	7.65	948.64	62.49	961.13	68.06	896.99	171.55
I1-31015	744.57	2.35	744.57	9.97	735.38	9.65	731.77	28.49
I1-31020	979.07	4.55	860.25	43.13	881.31	111.61	851.18	189.97
I1-31025	1131.59	3.94	1105.91	28.03	1122.54	72.05	1105.91	113.48
I1-41020	1287.14	9.49	1258.91	69.53	1224.35	13.14	1158.92	243.36
I1-41025	1588.95	45.01	1588.95	64.07	1588.95	72.15	1582.01	308.68
I1-5850	1236.65	15.57	1226.24	100.34	1252.40	231.45	1210.27	521.72
I1-51050	1256.59	30.26	1300.78	274.83	1280.79	439.94	1279.02	853.57
I1-51075	1669.67	61.06	1679.84	342.48	1649.91	429.38	1591.60	1026.12
I1-51575	1780.32	32.82	1739.81	204.83	1754.44	828.39	1708.79	2614.13
I1-510100	2458.50	121.66	2392.56	558.87	2401.88	621.73	2257.35	1906.17
I1-520100	2124.69	249.70	2087.29	1340.48	2089.68	2517.46	2071.76	3780.61
I1-510150	2220.47	345.53	2105.09	1115.53	2095.48	1662.28	2097.81	3740.38
I1-520150	2098.87	538.99	2076.23	2243.41	2029.99	2858.92	1919.35	3271.92
I1-510200	2761.73	440.27	2708.19	1697.46	2751.71	1944.08	2601.33	2239.09
I1-520200	2546.74	473.41	2457.04	3125.20	2446.01	3844.64	2407.33	6037.38

Table 6.5: Experimental results of TS settings on test instances I1.

Instance	TS1	CPU-1	TS2	CPU-2	TS3	CPU-3	TS4	CPU-4
I2-238	589.38	0.42	589.38	0.45	589.38	0.42	589.38	0.66
I2-239	413.54	0.54	413.54	0.49	413.54	0.49	413.54	1.01
I2-248	605.40	0.56	605.40	0.69	605.40	0.58	605.40	1.61
I2-2410	629.38	0.91	629.38	1.17	629.38	1.26	629.38	2.34
I2-2415	943.35	1.76	938.04	1.59	947.41	1.40	912.73	3.71
I2-3510	551.45	1.13	551.45	2.74	551.45	1.06	551.45	5.87
I2-3515	1214.31	6.05	1210.45	9.98	1201.30	6.05	1170.83	32.12
I2-2820	867.41	2.67	829.82	20.14	842.02	32.00	822.85	37.72
I2-2825	959.13	6.07	993.02	20.98	955.03	29.17	956.34	37.48
I2-21015	749.17	3.97	741.73	18.33	731.54	30.50	727.77	39.36
I2-21020	856.57	4.17	813.80	21.57	813.80	40.75	790.57	54.55
I2-21025	1017.53	4.79	1052.18	14.17	1012.04	52.19	961.74	61.29
I2-3810	583.73	1.06	504.20	8.51	504.20	11.35	504.20	13.04
I2-3815	688.68	1.74	672.42	5.56	688.68	15.06	685.48	20.06
I2-3820	769.04	4.57	769.04	32.35	762.25	42.29	765.01	82.03
I2-3825	1055.80	4.73	1069.02	20.69	1037.59	52.89	1026.36	38.65
I2-31015	813.52	2.15	791.13	27.01	791.13	73.87	777.49	82.22
I2-31020	843.23	5.39	827.15	30.14	821.75	134.35	794.58	153.01
I2-31025	1015.10	6.56	1021.21	43.59	1013.32	71.00	1010.51	152.30
I2-41020	868.03	20.29	868.03	53.04	856.40	9.57	802.60	433.90
I2-41025	1193.23	20.35	1193.23	43.31	1193.23	10.50	1185.31	320.03
I2-5850	1207.39	21.02	1185.75	198.69	1180.46	145.55	1185.75	665.22
I2-51050	1350.55	18.08	1348.33	157.06	1133.05	210.83	1335.81	390.62
I2-51075	1813.01	68.34	1784.81	370.43	1772.13	525.61	1756.88	1252.88
I2-51575	1710.38	53.43	1843.75	280.29	1809.01	568.79	1644.79	944.35
I2-510100	2411.03	60.60	2320.13	381.25	2299.03	265.96	2290.64	769.24
I2-520100	2051.39	257.63	2078.37	817.05	2049.21	2053.00	2041.13	2608.40
I2-510150	2018.49	302.78	1937.35	1674.89	1931.36	1171.31	1907.71	4852.92
I2-520150	1772.90	631.48	1764.34	2650.43	1806.03	3618.01	1707.73	4540.74
I2-510200	2435.05	101.01	2522.22	476.34	2542.03	797.35	2407.88	1078.87
I2-520200	2260.65	1237.99	2343.11	3656.93	2265.41	5921.24	2223.72	7850.52

Table 6.6: Experimental results of TS settings on test instances I2.

Instances	TS1	CPU-1	TS2	CPU-2	TS3	CPU-3	TS4	CPU-4
I3-238	589.80	0.39	589.80	0.38	589.78	0.92	589.78	1.00
I3-239	466.01	0.41	466.01	0.41	486.08	0.48	454.63	1.01
I3-248	451.62	1.61	451.62	0.89	504.70	0.92	451.62	1.61
I3-2410	546.36	0.92	546.36	0.97	546.36	1.08	546.36	2.29
I3-2415	805.46	1.25	805.46	1.47	787.45	2.18	718.16	4.73
I3-3510	747.37	1.76	747.37	1.61	747.37	2.86	745.85	5.49
I3-3515	1071.98	2.48	1054.23	3.42	1071.98	3.61	1033.79	7.08
I3-2820	893.36	2.59	892.21	20.98	858.09	11.44	829.20	32.24
I3-2825	1004.86	4.89	977.81	29.29	967.11	16.24	959.97	41.69
I3-21015	620.86	1.98	620.86	13.39	620.86	5.86	620.86	15.05
I3-21020	757.21	2.54	757.21	28.10	756.71	8.22	756.51	38.06
I3-21025	879.83	5.74	895.54	53.61	879.83	21.34	867.60	49.18
I3-3810	490.78	1.00	490.78	9.32	490.78	2.95	412.91	14.88
I3-3815	626.84	1.39	637.22	18.41	624.55	14.45	624.55	22.35
I3-3820	732.83	3.22	710.09	110.21	732.63	37.98	707.57	187.18
I3-3825	860.26	4.10	833.43	71.69	830.26	29.28	806.71	133.70
I3-31015	624.73	1.70	586.94	19.66	613.72	18.34	574.26	40.61
I3-31020	781.39	3.69	775.29	137.71	770.77	30.63	745.85	256.91
I3-31025	913.31	2.70	897.71	126.28	891.83	21.48	860.81	91.11
I3-41020	1301.56	10.49	1216.79	7.58	1234.44	79.17	1204.57	274.56
I3-41025	1141.80	20.28	1125.74	29.49	1110.40	55.73	1089.40	467.69
I3-5850	1351.27	16.67	1292.54	75.23	1233.59	107.61	1240.80	474.94
I3-51050	1297.51	24.68	1256.68	145.49	1253.78	109.82	1243.87	919.53
I3-51075	1937.27	45.20	1911.85	293.33	1881.52	314.64	1839.38	806.94
I3-51575	1602.72	42.19	1653.44	299.98	1635.61	875.87	1590.00	1910.94
I3-510100	2420.47	37.79	2366.36	272.50	2323.13	329.78	2294.44	546.61
I3-520100	2278.57	75.72	2197.11	283.22	2185.37	431.00	2170.45	696.22
I3-510150	1398.69	182.22	1387.54	1132.39	1414.37	1067.02	1342.18	2635.06
I3-520150	1454.31	232.34	1357.09	862.33	1382.10	2150.06	1343.72	3379.30
I3-510200	2030.30	351.31	1910.32	979.62	1914.98	2194.54	1893.68	2633.48
I3-520200	2737.23	343.97	2728.95	1081.43	2706.50	2375.41	2692.31	2765.42

Table 6.7: Experimental results of TS settings on test instances I3.

	TS1	CPU-1	DTS1	CPU-DTS1
I1-5850	1236.65	15.57	1236.65	50.565
I1-51050	1256.59	30.26	1271.1	90.93
I1-51075	1669.67	61.06	1669.67	196.0518
I1-51575	1780.32	32.82	1804.57	240.3384
I1-510100	2458.50	121.66	2407.98	443.6748
I1-520100	2124.69	249.70	2105.5	697.62
I1-510150	2220.47	345.53	2215.24	649.98
I1-520150	2098.87	538.99	2056.64	1511.056
I1-510200	2761.73	440.27	2761.73	855.624
I1-520200	2546.74	473.41	2457.99	2702.682
I2-5850	1207.39	21.02	1232.38	96.24
I2-51050	1350.55	18.08	1403.55	64.107
I2-51075	1813.01	68.34	1813.01	234.198
I2-51575	1710.38	53.43	1843.75	259.6272
I2-510100	2411.03	60.60	2415.22	194.448
I2-520100	2051.39	257.63	2174.98	487.257
I2-510150	2018.49	302.78	1968.89	781.32
I2-520150	1772.90	631.48	1735.57	1208.88
I2-510200	2435.05	101.01	2571.76	317.442
I2-520200	2260.65	1237.99	2330.65	3809.082
I3-5850	1351.27	16.67	1254.43	75.6462
I3-51050	1297.51	24.68	1291.18	57.93942
I3-51075	1937.27	45.20	1907.59	154.17
I3-51575	1602.72	42.19	1598.5	124.0308
I3-510100	2420.47	37.79	2383.76	125.2008
I3-520100	2278.57	75.72	2255.95	203.4492
I3-510150	1398.69	182.22	1371.78	429.06
I3-520150	1454.31	232.34	1424.67	790.8
I3-510200	2030.30	351.31	1939.23	765.102
I3-520200	2737.23	343.97	2814.32	695.52

Table 6.8: Results of diversification for TS setting 1.

From tables 6.8 and 6.9 we observe that the used diversification criterion allows to find better solutions in most of the instances for *setting 1*, whereas for *setting 4* the improvements are lower and in several instances no better solutions are determined. On the other side, in the most of the cases, computation times can increase in a relevant way.

6.2 Comparisons of TS with exact methods

In this section the results of Tabu Search are compared with the solutions obtained with the commercial solver and reported in *Chapter 5*, in order to evaluated the effectiveness of our TS method in terms of quality of solution and computation times with reference to the available bounds.

For small instances we compare the TS results with the best result

	TS4	CPU-4	DTS4	CPU-DTS4
I1-5850	1210.27	521.72	1210.27	658.482
I1-51050	1279.02	853.57	1279.02	958.482
I1-51075	1591.60	1026.12	1571.83	1470.378
I1-51575	1708.79	2614.13	1682.21	3105.36
I1-510100	2257.35	1906.17	2375.32	2016.384
I1-520100	2071.76	3780.61	2071.76	4887.9
I1-510150	2097.81	3740.38	2116.79	4149.3
I1-520150	1919.35	3271.92	1913.23	4561.56
I1-510200	2601.33	2239.09	2601.33	2615.57
I1-520200	2407.33	6037.38	2404.38	7712.04
I2-5850	1185.75	665.22	1185.75	774.273
I2-51050	1335.81	390.62	1337.39	413.796
I2-51075	1756.88	1252.88	1694.75	2062.89
I2-51575	1644.79	944.35	1623.36	1107.072
I2-510100	2290.64	769.24	2323.38	705.03
I2-520100	2041.13	2608.40	2041.13	2711.21
I2-510150	1907.71	4852.92	1916.79	4523.67
I2-520150	1707.73	4540.74	1758	6374.94
I2-510200	2407.88	1078.87	2407.88	1387.54
I2-520200	2223.72	7850.52	2211.85	8101.23
I3-5850	1240.80	474.94	1248.69	427.42
I3-51050	1243.87	919.53	1250.17	596.8482
I3-51075	1839.38	806.94	1863.23	1470.378
I3-51575	1590.00	1910.94	1597.63	2505.36
I3-510100	2294.44	546.61	2328.17	1296.384
I3-520100	2170.45	696.22	2166.36	1227.9
I3-510150	1342.18	2635.06	1336.61	2949.3
I3-520150	1343.72	3379.30	1372.17	3012.63
I3-510200	1893.68	2633.48	1915.45	2478.12
I3-520200	2692.31	2765.42	2673.15	3285.45

Table 6.9: Results of diversification for TS setting 4.

obtained with one of the two formulations (three-index and assignment based) and the related computation times. Whereas for medium and large instances the results are compared with the ones obtained with the decomposition approach.

The evaluation of the gap $\Delta(z)$ between TS and bounds, for a generic instance I, is computed with the following expression:

$$\Delta(z) = [1 - z(TS_I)/z(BS_I)] \tag{6.1}$$

where $z(TS_I)$ and $z(BS_I)$ are respectively the solutions obtained with the TS heuristic and by the commercial solver for the same instance I. A positive value indicates that TS solution improves available bound, whereas negative values indicate that bound value is better than the solution provided by TS.

Results on small instances are reported in tables 6.10, 6.11, 6.12. From these tables we can observe that in all cases where the optimal solution for the instance was known, TS has been able to determine it at least with one setting. More precisely, concerning *setting 1* of TS, the gap varies between $+0.206$ and -0.208. In the worst cases it is equal to -0.208 for set *I1*, -0.158 for set *I2* and -0.189 for set *I3*. On the other side computation time are always lower than *45* seconds. Concerning instead *setting 4* of TS, the gap is in the most of the cases positive and it varies between $+0.256$ and -0.051. Computation times increase with reference to *setting 1*, but they are significantly lower than the ones of the solver (less than *360* seconds). As explained in the appendix, the three sets of instances I1, I2 and I3, differ for the spatial distribution of the satellites. We can observe that the results of the TS on small instances seem to be not affected by this distribution.

In tables 6.13, 6.14 and 6.15 results on medium and large size instances are reported. Concerning *setting 1* we can observe that TS results are very close to the ones of the decomposition approach, but the saving in terms of computation time is meaningful. The gap varies between $+0.283$ and -0.094 and computation times are always lower than *600* seconds with the only exception of instance *I2-41025*. Concerning instead *setting 4*, it outperforms decomposition approach in most of the instances. The saving in terms of computation time is not so large as for *setting 1*, but for several instances the positive gap between the solutions is relevant. In particular the gap varies between $+0.295$ and -0.008 and computation time varies between $390,62$ and 7850.52 seconds.

Instance	BS	CPU	TS1	CPU-1	GAP-1	TS4	CPU-4	GAP-4
I1-238	591.83	10.23	591.83	0.39	0.000	591.83	0.50	0.000
I1-239	878.69	9.87	902.45	0.59	-0.027	878.69	1.08	0.000
I1-248	625.96	175.60	625.96	0.85	0.000	625.96	1.67	0.000
I1-2410	862.91	582.90	862.91	0.85	0.000	862.91	4.07	0.000
I1-2415	1105.67	1469.90	1121.50	1.92	-0.014	1105.67	4.15	0.000
I1-3510	829.25	2194.70	952.86	1.25	-0.149	829.25	5.29	0.000
I1-3515	1019.57	3893.50	1068.00	2.21	-0.048	1019.57	6.16	0.000
I1-2820	1055.65	7200.00	1114.41	5.28	-0.056	1055.20	48.22	0.000
I1-2825	992.08	7200.00	1021.69	4.00	-0.030	979.85	35.91	0.012
I1-21015	732.48	7200.00	754.63	1.53	-0.030	732.48	10.81	0.000
I1-21020	951.01	7200.00	1008.17	3.40	-0.060	947.65	51.94	0.004
I1-21025	1170.72	7200.00	1085.67	6.81	0.073	1084.26	86.17	0.074
I1-3810	604.37	4982.30	604.37	1.24	0.000	604.37	11.26	0.000
I1-3815	730.36	7200.00	730.36	1.61	0.000	730.36	11.12	0.000
I1-3820	898.75	7200.00	968.59	5.54	-0.078	898.08	154.62	0.001
I1-3825	1141.26	7200.00	943.25	7.65	0.173	896.99	171.55	0.214
I1-31015	699.11	7200.00	744.57	2.35	-0.065	731.77	28.49	-0.047
I1-31020	810.26	7200.00	979.07	4.55	-0.208	851.18	189.97	-0.051
I1-31025	1291.68	7200.00	1131.59	3.94	0.124	1105.91	113.48	0.144
I1-41020	1208.72	7200.00	1287.14	9.49	-0.065	1158.92	243.36	0.041
I1-41025	1615.33	7200.00	1588.95	45.01	0.016	1582.01	308.68	0.021

Table 6.10: Tabu Search vs. models on small instances I1.

Instance	BS	CPU	TS1	CPU-1	GAP-1	TS4	CPU-4	GAP-4
I2-238	589.38	6.45	589.38	0.42	0.000	589.38	0.66	0.000
I2-239	413.54	8.31	413.54	0.54	0.000	413.54	1.01	0.000
I2-248	605.40	182.50	605.40	0.56	0.000	605.40	1.61	0.000
I2-2410	629.38	834.30	629.38	0.91	0.000	629.38	2.34	0.000
I2-2415	912.73	1525.30	943.35	1.76	-0.034	912.73	3.71	0.000
I2-3510	551.45	2281.50	551.45	1.13	0.000	551.45	5.87	0.000
I2-3515	1170.83	4365.50	1214.31	6.05	-0.037	1170.83	32.12	0.000
I2-2820	822.85	7200.00	867.41	2.67	-0.054	822.85	37.72	0.000
I2-2825	947.84	7200.00	959.13	6.07	-0.012	956.34	37.48	-0.009
I2-21015	727.77	7200.00	749.17	3.97	-0.029	727.77	39.36	0.000
I2-21020	801.28	7200.00	856.57	4.17	-0.069	790.57	54.55	0.013
I2-21025	1263.54	7200.00	1017.53	4.79	0.195	961.74	61.29	0.239
I2-3810	504.20	6412.23	583.73	1.06	-0.158	504.20	13.04	0.000
I2-3815	685.48	7200.00	688.68	1.74	-0.005	685.48	20.06	0.000
I2-3820	805.38	7200.00	769.04	4.57	0.045	765.01	82.03	0.050
I2-3825	1026.36	7200.00	1055.80	4.73	-0.029	1026.36	38.65	0.000
I2-31015	812.13	7200.00	813.52	2.15	-0.002	777.49	82.22	0.043
I2-31020	806.67	7200.00	843.23	5.39	-0.045	794.58	153.01	0.015
I2-31025	1254.62	7200.00	1015.10	6.56	0.191	1010.51	152.30	0.195
I2-41020	1093.34	7200.00	868.03	20.29	0.206	802.60	433.90	0.266
I2-41025	1380.86	7200.00	1193.23	20.35	0.136	1185.31	320.03	0.142

Table 6.11: Tabu Search vs. models on small instances I2.

Instance	BS	CPU	TS1	CPU-1	GAP-1	TS4	CPU-4	GAP-4
I3-238	589.80	8.13	589.80	0.39	0.000	589.78	1.00	0.000
I3-239	454.63	7.10	466.01	0.41	-0.025	454.63	1.01	0.000
I3-248	451.62	164.70	451.62	1.61	0.000	451.62	1.61	0.000
I3-2410	546.36	416.80	546.36	0.92	0.000	546.36	2.29	0.000
I3-2415	718.16	1225.30	805.46	1.25	-0.122	718.16	4.73	0.000
I3-3510	745.85	2674.20	747.37	1.76	-0.002	745.85	5.49	0.000
I3-3515	1033.79	3065.50	1071.98	2.48	-0.037	1033.79	7.08	0.000
I3-2820	829.20	7200.00	893.36	2.59	-0.077	829.20	32.24	0.000
I3-2825	1100.31	7200.00	1004.86	4.89	0.087	959.97	41.69	0.128
I3-21015	620.86	7200.00	620.86	1.98	0.000	620.86	15.05	0.000
I3-21020	790.99	7200.00	757.21	2.54	0.043	756.51	38.06	0.044
I3-21025	944.84	7200.00	879.83	5.74	0.069	867.60	49.18	0.082
I3-3810	412.91	3376.23	490.78	1.00	-0.189	412.91	14.88	0.000
I3-3815	624.55	7200.00	626.84	1.39	-0.004	624.55	22.35	0.000
I3-3820	707.57	7200.00	732.83	3.22	-0.036	707.57	187.18	0.000
I3-3825	977.10	7200.00	860.26	4.10	0.120	806.71	133.70	0.174
I3-31015	574.26	7200.00	624.73	1.70	-0.088	574.26	40.61	0.000
I3-31020	789.49	7200.00	781.39	3.69	0.010	745.85	256.91	0.055
I3-31025	1038.58	7200.00	913.31	2.70	0.121	860.805	91.11	0.171
I3-41020	1287.23	7200.00	1301.56	10.49	-0.011	1204.57	274.56	0.064
I3-41025	1089.40	7200.00	1141.80	20.28	-0.048	1089.40	467.69	0.000

Table 6.12: Tabu Search vs. models on small instances I3.

Instance	DA-BS	CPU	TS1	CPU-1	GAP-1	TS4	CPU-4	GAP-4
I1-5850	1226.24	4421.30	1236.65	15.57	-0.008	1210.27	521.72	0.013
I1-51050	1783.60	6134.90	1279.02	30.26	0.283	1256.59	853.57	0.295
I1-51075	1591.60	7512.60	1669.67	61.06	-0.049	1591.60	1026.12	0.000
I1-51575	1783.60	6134.90	1780.32	32.82	0.002	1708.79	2614.13	0.042
I1-510100	2247.32	8033.80	2458.50	121.66	-0.094	2257.35	1906.17	-0.004
I1-520100	2055.88	10218.10	2124.69	249.70	-0.033	2071.76	3780.61	-0.008
I1-510150	2177.77	8407.10	2220.47	345.53	-0.020	2097.81	3740.38	0.037
I1-520150	1933.82	7786.60	2098.87	538.99	-0.085	1919.35	3271.92	0.007
I1-510200	2625.11	10119.50	2761.73	440.27	-0.052	2601.33	2239.09	0.009
I1-520200	3140.17	12750.30	2546.74	473.41	0.189	2407.33	6037.38	0.233

Table 6.13: Tabu Search vs. decomposition approach on medium-large instances I1.

Instance	DA-BS	CPU	TS1	CPU-1	GAP-1	TS4	CPU-4	GAP-4
I2-5850	1185.75	2023.34	1207.39	21.02	-0.018	1185.75	665.22	0.000
I2-51050	1325.61	5039.50	1350.55	18.08	-0.019	1335.81	390.624	-0.008
I2-51075	1768.88	7061.00	1813.01	68.34	-0.025	1756.13	1252.878	0.007
I2-51575	1644.79	9499.40	1710.38	53.43	-0.040	1644.79	944.352	0.000
I2-510100	2391.17	10379.60	2411.03	60.60	-0.008	2290.64	769.242	0.042
I2-520100	2051.39	12405.60	2051.39	257.63	0.000	2041.13	2608.404	0.005
I2-510150	2111.97	14060.90	2018.49	302.78	0.044	1907.71	4852.92	0.097
I2-520150	1800.89	10134.50	1772.90	631.48	0.016	1707.73	4540.74	0.052
I2-510200	2430.93	8871.80	2435.05	101.01	-0.002	2407.88	1078.866	0.009
I2-520200	2274.29	15602.10	2260.65	1237.99	0.006	2223.72	7850.52	0.022

Table 6.14: Tabu Search vs. decomposition approach on medium-large instances I2.

Instance	DA-BS	CPU	TS1	CPU-1	GAP-1	TS4	CPU-4	GAP-4
I3-5850	1298.89	7741.90	1351.27	16.67	-0.040	1240.80	474.94	0.045
I3-51050	1256.68	4929.60	1297.51	24.68	-0.032	1243.87	919.53	0.010
I3-51075	1879.56	13720.00	1937.27	45.20	-0.031	1839.38	806.94	0.021
I3-51575	1704.65	12903.90	1602.72	42.19	0.060	1590.00	1910.94	0.067
I3-510100	2601.44	20599.60	2420.47	37.79	0.070	2294.44	546.61	0.118
I3-520100	2261.36	15724.50	2278.57	75.72	-0.008	2170.45	696.22	0.040
I3-510150	1470.77	243.90	1398.69	182.22	0.049	1342.18	2635.06	0.087
I3-520150	1508.07	21240.50	1454.31	232.34	0.036	1343.72	3379.30	0.109
I3-510200	2193.32	41145.10	2030.30	351.31	0.074	1893.68	2633.48	0.137
I3-520200	2784.47	23319.40	2737.23	343.97	0.017	2692.31	2765.42	0.033

Table 6.15: Tabu Search vs. decomposition approach on medium-large instances I3.

To summarize TS results well compare with the available bounds for all the 2E-LRP generated instances. The spatial distribution of the secondary facilities does not affect the quality of the solutions but it can affect the related computation times.

Part III

Flow intercepting facility location (FIFLP): problems, models and heuristics

Chapter 7

Problem definition and related models

In this chapter flow interception facility location problem (FIFLP) are presented. The chapter starts with a description of the FIFLP and its main application fields. Then it focuses on the literature review, classified in function of: definition of general models and methods, application to traffic and transportation problems, application to communication network problems. Then a discussion on five key issues in flow interception problem definition is provided. Four fixed flow intercepting facility location problems are treated. Each of them has been formulated as a mixed-integer model, which differs for the functions defined on the path to intercept. The chapter concludes with a presentation of several modifications of the proposed models and with an adaptation of them to the mobile facility case.

7.1 Introduction to FIFLP

The network location literature, starting from the seminal papers of Hakimi [50, 51], is very broad and mainly devoted to the location of plants, facilities and services which perform activities producing the generation and/or the attraction of people, goods, materials, energy, information, so generating and/or attracting customer flows. The flows of customers reach the plants or, vice versa, the plants generate flows which reach the customers.

In the last 25 years, from the paper of Hodgson [53], a significant number of papers have treated the interesting case of the location of facilities which do not generate and/or attract flow, but intercept it. In any case these facilities (in the following also referred as devices depending on the application field) perform activities which can be exploited by the flow units of the network or proposed to/imposed on them along their pre-planned path from an origin to a destination. In other words the purpose of the flow units is not to obtain a service, but, if there is a facility on their pre-planned path, they may voluntarily or obligatorily interrupt the movement to obtain the service, before continuing their path. For this reason the expressions "flow interception" or "flow intercepting facility location problems" (FIFLP) are generally adopted. The FIFLP has been approached both in deterministic and stochastic scenarios. In the former case we assume that there is a complete knowledge of all the paths that carry non-zero flows, which are also assumed to be known. In the latter case the information about paths and flows are not available, but the information about the fraction of flows that travel from any node to all adjacent nodes are known.

This chapter is focused on the FIFLP in a deterministic scenario. To this aim at first a wide literature review on the subject is provided, with a presentation of the main contributions in transportation and communication network fields. Then a preliminary discussion is presented about five issues, relevant for an unambiguous definition of the flow intercepting problems.

7.2 Literature review

The literature review on the FIFLP is aimed to show that these problems and their variants arise in different fields, confirming that the problem is a living matter. The literature on FIFLP is really heterogeneous, but it concerns basically three main class of issues [16]:

- definition of general models and methods,
- application to traffic and transportation problems,
- application to communication problems.

- *Definition of general models and methods*

With reference to this issue, many works were proposed by Berman et al., starting from the beginnings of the nineties.

Berman et al. [13] propose a model and a greedy heuristic for the problem of the optimal location of "discretionary facilities" (i.e. for example automatic teller machine and gasoline station) on a network. They also propose an inverse formulation for this problem and a heuristic able to solve the problem to the optimum.

Berman et al. [10] present some generalizations of the models for the optimal location of discretionary facilities, all sharing the property that customer flows may deviate from pre-planned paths in order to visit a facility. They formalize three problems, two of which can be solved by greedy heuristics and the third by any approximate or exact method able to solve a p-median problem.

Berman et al. [12] study the problem of locating flow intercepting facilities with probabilistic flows. They formulate two non linear integer programming models, for the single and the double counting cases. They also derive a linear integer program for these two models and propose a simple greedy heuristic for them.

Berman [8] treats the problem of locating discretionary facilities with finite capacities on a network, where customers can deviate from their pre-planned trip. He presents a scheme able to calculate the expected number of customers who travel to a single facility. He also provides a location-allocation heuristic algorithm for the case of locating more than one facility.

Averbakh and Berman [4] formulate two integer programming models for the problem of locating flow capturing facilities on a transportation network, where the level of customer usage of a service depends on the number of facilities that they encounter on their path. They propose a heuristic for one of the problems and a polynomial algorithm for the other one in case the network is a tree.

Berman [9] combines the models of flow interception and the traditional location models. He proposes four new problems and shows that they have a structure similar to that of other known location problems.

Berman and Krass [11] propose a model for the location of competitive facilities, combining the features of spatial interaction and flow interception models. They propose a heuristic and a branch and bound scheme for the model.

- *Application to traffic and transportation problems*

Applications to transportation networks can be grouped in three main categories, depending on the specific type of facility we want to locate:

- location of traffic counting sensors for origin-destination (o-d) matrix estimation problems,
- location of inspection stations for the hazardous material transportation or for control problems,
- location of Variable Message Signs (VMS) for route guidance.

With reference to the first category, Yang and Zhou [109] define some rules to locate sensors on arcs to better estimate the o-d matrix and propose a greedy heuristic described in detail in the following, to determine the counting links satisfying four location rules. Bianco et al. [15] approach the sensor location problem using an objective function aimed to maximize the likelihood of the o-d matrix estimated through the counted flows. Tomas [95] uses a Constraint Logic Programming approach to find the minimum number and location of traffic counters to obtain o-d data. Yang et al. [107] include the planning horizon in the traffic counting location problem, to optimize not only the location of the sensors, but also the time of the implementation. The objective is to maximize the number of o-d pairs covered in each year, with an additional budget constraint. The problem is formulated as an integer-programming model solved by a genetic algorithm. Gentili and Mirchandani [44] introduce new network location problems to determine where "active sensors" have to be located to monitor or manage particular traffic streams. Yang et al. [108] consider the problem of the optimal selection of screen lines for traffic census in road networks from two view points: to find the optimal location of a given number of counting stations to separate as many o-d pairs as possible, or to determine the minimum number of counting stations and their location, required to separate all o-d pairs. An o-d pair is defined "separated" if its flows are entirely intercepted by the traffic counting stations. The problems are formulated as integer linear programming models solved by branch and bound techniques.

With reference to the category of inspection stations, Mirchandani et al. [78] approach the problem with reference to hazardous material transportation. They described some heuristics and compute bounds for them. Hodgson et al. [54] develop a new conceptual approach to locate inspection stations for hazardous vehicles, in terms of preventive, instead of punitive, interception. Rosenkrantz et al. [86] approach two categories of problems for transportation and communication networks: location of inspection stations along a path from an origin to a destination and simultaneous selections of a path and inspection stations along this path. Gendreau et al. [43] describe formulations and properties for the punitive and preventive flow interception problem devoted to solving the inspection station location, and propose some heuristics for the preventive case, in order to maximize the total risk reduction.

The third category (VMS location) is approached by Huynh et al. [56]. They address the problem of finding the best location for portable variable message signs to divert traffic to alternate paths when an incident occurs, so that the incident's impact on the network is minimized.

- *Application to communication problems*

The flow interception problem arises in the context of communication networks with the aim of flow monitoring and control. It consists in the use of several monitoring devices (monitors or probes) which, placed inside the routers or deployed as a standalone box on the links of a communication network, summarize and record information about traffic flows. This problem, approached with various covering and location models, has never been treated explicitly, to the best of our knowledge, as a problem of flow interception on the network, between origin and destination nodes.

Chaudet et al. [22, 23] treat the monitoring problem in two cases, assigning tap devices for passive monitoring and assigning beacons for active monitoring. They propose a Mixed Integer Linear Programming formulation, derived from a minimum edge cost flow model, and a greedy algorithm for the determination of the number of devices to locate, minimizing set up cost or set up and deployment cost, with or without sampling. They also approach the problem of finding the minimum number of beacons (i.e. nodes in charge of the monitoring task and emitting packets) whose probes (i.e. packets emitted by the beacons) cover all the links of the network, solving it to the optimum by an integer linear programming model or by a greedy heuristic.

Cantieni et al. [20], instead, examine the device placement problem from a different point of view. They suppose that all links can be monitored and so they determine which devices have to be activated and which sampling rate should be set on them in order to achieve a given measurement task with high accuracy and low resource consumption. They formulate the problem as a non linear constrained model solved by a Lagrangean multiplier method and by a gradient projection method.

Hu et al. [55] formulate the passive monitoring problem as a Stochastic Constrained Optimization model. They propose an algorithm that returns the optimal placement for the devices of a distributed passive measurement system and their sampling rate, in order to maximize the probability of a packet being sampled.

Suh et al. [91] propose models of budget constrained maximum coverage to find the optimal location of devices in order to maximize a utility function, which expresses the benefit achievable from flow monitoring activity, with or without sampling. They also propose inverse models to minimize deployment and management costs, constrained to obtain a prefixed value of the achievable benefit. They also present models related to the case of link failure. The models have been solved to the optimum and moreover by the greedy heuristic of Khuller et al. [59] with some modifications.

7.3 Five key issues in problem definition

The flow intercepting facility location problems, treated with a deterministic approach, require a preliminary discussion of the following five key issues:

- knowledge of the origin-destination paths and related flows,
- link failures and/or flow deviations from predefined paths,
- single or multiple interception of flows, i.e. single or multiple evaluations and/or counting of them,
- flow monitoring with or without sampling operations,
- location in nodes or on links of the network.

- *Knowledge of the origin-destination paths and related flows*

Models under investigation need information about all used paths and relative flows for each o-d pair on the network. If this information is not available, we could obtain it through the knowledge of the traffic demand matrix. The flow related to each o-d pair can be assigned to a single path, generally the shortest one, or to a set of paths, if we are operating in a context of load balancing and flow equilibrium. If the traffic demand matrix is not available, we can obtain it by an o-d matrix estimation method ([7],[21]) or by the random generation of the o-d flows, if this truly represents the real network behavior. In this paper we consider the following assumption with reference to the knowledge of the o-d paths:

Assumption 1: we have a complete knowledge of all paths that carry non-zero flows, which are also assumed to be known. Each o-d flow is assumed using the shortest path. For this reason, in the following, we will speak in terms of set of paths and not in terms of o-d pairs.

- *Link failure and/or flow deviation from predefined paths*

There may be two possible situations. If no link failures occur, flow deviations from predefined paths are not possible, otherwise it is necessary to define an alternative for each path using that link. In this paper we consider the following assumption:

Assumption 2: no link failure occurs and so flow deviation from predefined paths is not possible.

- *Single or multiple interception of flows, i.e. single or multiple evaluation and/or counting of them*

This issue underlines the fact that a flow can encounter more than one facility along its path, but it is not necessarily intercepted by more of them. The choice between single or multiple interception has to be based on the specific aim of the problem, which could require that each flow is evaluated once or more times in the computation of the objective function. In this paper we consider the following assumption:

Assumption 3: Multiple interception (evaluation) of flows will not be allowed in the objective function.

- *Flow monitoring with or without sampling operations*

There are two possible strategies usable for the FIFLP, i.e. each flow is entirely intercepted or partially intercepted, according to a sampling operation, by a facility. Moreover we note that in the second case, each flow can be intercepted more than once with different sampling rates and, on the other hand, each facility can intercept different flows with different sampling rates. In this paper we consider the following assumption:

Assumption 4: Each intercepted flow is entirely monitored by a single facility.

- *Location in nodes or on links of the network*

This issue concerns the possibility of locating facilities in the nodes or on the links of the network, with reference to the characteristics of the network system. In order to explain the difference between the two alternatives, let us consider a network *G(N,A)*, where N is the set of nodes and A is the set of links. Moreover let P be the set of all paths that carry non-zero flows. Let us hypothesize that we are interested in locating m devices with the aim of maximizing the intercepted flow. Let S be a set of m possible points of G and let x_{sp} be a binary variable which assumes value *1* if a path p is intercepted by at least one point in S, else *0*. If f_p is the flow value associated to path p, the flow intercepted by the points of S can be written as $z(S) = \sum_{p \in P} f_p x_{sp}$. So the problem can be formalized as follows:

$$Max\ z(S), S \subseteq G, |S| = m$$

We can assert that the selection of the m solution points can be limited to the nodes of set N and therefore all inner points between two nodes can be excluded. This proposition can be informally demonstrated ([13]) noting that a facility, placed on a link, can be moved to one of its two extreme nodes not only without any loss of intercepted flow, but also with a possible increase of an additional flow, traversing another link incident in the same node. We can therefore say that in the maximization of function *z(S)*, S can be defined as a subset of N and therefore the problem becomes:

$$Max\ z(S), S \subseteq N, |S| = m$$

Consequently when it is possible, node-location has to be preferred to link-location. For this reason in the following all the models will be shown in the node-location formulation, remembering that, whenever link-location formulation is better for the decisional problem, it can be easily obtained by replacing node variables with link variables in the objective function and in the constraints of the models. In this paper we consider the following assumption:

Assumption 5: facilities are located in nodes.

7.4 FIFLP formulations

In this chapter several formulations for the FIFLP are presented. The models are basically path-covering models which differ for the kind of used variables and for the kind of facilities to locate on the network (fixed or mobile). The difference is in the fact that fixed facility are located on the network and cannot be moved after installation, whereas mobile facilities can be relocated on the network in function of the status of the network, The main focus is on fixed facility location problems, which can be classified in two categories: flow oriented problems and gain oriented problems. The first category problems adopt a performance criterion or a constraint which are directly related to the flow value on each path, whereas the second category problems adopt a new performance criterion aimed to maximize an obtainable "gain" and a constraint on the amount of the "gain" to achieve, which are implicitly related to the flow values. Therefore the chapter is structured as follows. At first several models for the flow oriented and gain oriented cases are provided. Then several extensions and specializations of these models are presented. To conclude a discussion about the adaptation of the proposed models to mobile facility location problem is provided.

7.5 Flow oriented problems

In the following, starting from the previous discussion, we will present two problems of flow intercepting facility location, which can be approached by binary linear programming models. The first model, which will be referred as the Maximum Flow Interception Model (*M1*), finds the optimal location of a limited number of facilities for the maximization of the intercepted flow (problem *P1*). The second one, the Minimum Number of Facilities for the Maximization of Intercepted Flow Model (*M2*), finds the minimum number and the location of the facilities needed to intercept a certain amount of flow on a network (problem *P2*). These models can be referred as “Flow Oriented” since they adopt respectively a performance criterion or a constraint which are directly related to the flow value on each path.

We will refer to a network *G(N,A)*, where $N = \{i\}$ is the set of nodes and $A = \{l\}$ is the set of links. Moreover we will define $P = \{p\}$ as the set of origin-destination paths selected on the network, N_p as the set of nodes belonging to a path p, $F = \{f_p\}$ as the set of flows on the paths belonging to P and m as the number of facilities to locate on the network. We need to define two kinds of decisional variables, node variables y_i and path variables x_p, which can assume the following values:

$y_i = \{0, 1\}$	*1*, if a facility is located at node i, *0* otherwise,
$x_p = \{0, 1\}$	*1*, if at least one facility is located on path p, *0* otherwise

7.5.1 Problem *P1*: maximization of the intercepted flow

Model M1 for problem P1 has been proposed by Berman et al. [13] and its formulation is similar to that proposed by Church and ReVelle [29] for the Maximal Covering Problem. It can be written as follows:

$$\text{Maximize} \quad \sum_{p \in P} f_p \, x_p$$

$$\text{s.t.} \quad \sum_{i \in V} y_i = m \tag{7.1}$$

$$\sum_{i \in N_p} y_i \geq x_p \qquad \forall p \in P \tag{7.2}$$

$$x_p = \{0, 1\} \qquad \forall p \in P$$

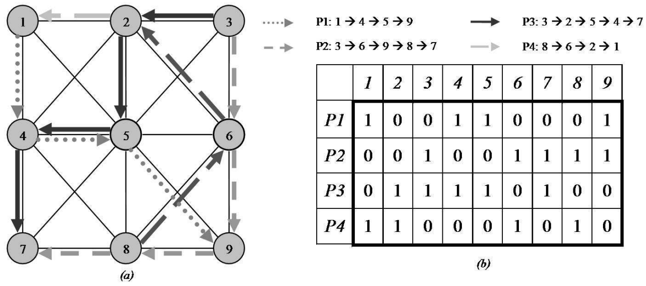
$$y_i = \{0, 1\} \qquad \forall i \in N$$

The objective function expresses the maximization of the intercepted flow. The constraint (1) imposes the number of facilities that have to be placed. Constraints (2) are consistency constraints between the two kinds of variables. Indeed, if no vertex of path p contains a facility (i.e. all y_i are equal to *0*, $i \in N_p$) the variable x_p must be equal to *0*, that is the path p is not intercepted. Otherwise, if at least one vertex of path p contains a facility (i.e. a variable y_i, $i \in N_p$, is equal to *1*) the variable x_p could be *0* or *1*. Because of the maximization of the objective function, the variable x_p assumes value *1*, that is the path p is intercepted. As stated above, even if a path flow encounters more than one facility, it will be intercepted and counted only once in the objective function (i.e. objective function formulation does not allow multiple counting).

The solution of this model returns the location of the m facilities giving the maximization of the intercepted flow. Note that if the flow on path p encounters more than one facility (i.e. more than one variable y_i, $i \in N_p$, assumes value *1*), we do not exactly know which is the one intercepting it. However, if this association between paths and facilities is needed, we can build a set of unambiguous correspondences between them, using a simple path covering heuristic based on the path-node incidence matrix, referred to as coverage matrix $\boldsymbol{B}$. In figure 7.1 *(a, b)* a simple network and the related path-node incidence matrix $\boldsymbol{B}$ are shown. The rows of matrix $\boldsymbol{B}$ are associate to the used o-d paths and the columns to the nodes of the network. Its generic element $b_{pi}(p \in P, i \in V)$ is equal to *1* if the node i belongs to the path p, *0* otherwise.

The path covering heuristic works in this way. Let S be the node solution set, sorted for increasing value of the index node. The first node i^* in the set is selected and it is assigned to all the paths associated to $b_{pi^ast} = 1$. This operation is repeated for all the nodes in S, assigning them to the remaining uncovered paths. This simple heuristic returns an unambiguous assignment of facilities and paths and moreover it allows to verify if the number of located facilities is redundant. In fact it can happen that a node j, $j \in S$, is not assigned to any path. Hence it is possible to eliminate it from the solution set without any decrease in the amount of intercepted flow.

In solving model *M1* it could also happen that if the number of facilities is high, some of them could be redundant, and therefore they provide no increases of the objective function, but additional costs. In order to avoid this situation we could substitute the constraint *(1)* with an inequality constraint, imposing that the number of facilities has to be less than or equal to m.



	1	2	3	4	5	6	7	8	9
P1	1	0	0	1	1	0	0	0	1
P2	0	0	1	0	0	1	1	1	1
P3	0	1	1	1	1	0	1	0	0
P4	1	1	0	0	0	1	0	1	0

(b)

Figure 7.1: Paths on a network and related path-node incidence matrix $\boldsymbol{B}$.

7.5.2 Problem *P2*: minimization of the number of flow intercepting facilities

Model M2 is aimed to minimize the number of facilities required to intercept an assigned percent of the total flow or the whole flow traversing the network. Let $(1-\epsilon)$ be the fraction of the total flow that has to be intercepted $(0 \leq \epsilon < 1)$ and let $K_\epsilon = (1-\epsilon)\sum_{p\in P} f_p$ be the corresponding flow value. The following binary linear programming model can be formulated:

$$\text{Minimize} \quad \sum_{i\in V} y_i$$

$$\text{s.t.} \quad \sum_{i\in N_p} y_i \geq x_p \qquad \forall p \in P \tag{2}$$

$$\sum_{p\in P} f_p\, x_p \geq K_\epsilon \tag{7.3}$$

$$x_p = \{0,1\} \qquad \forall p \in P$$

$$y_i = \{0,1\} \qquad \forall i \in N$$

The objective function expresses the minimization of the number of facilities. The set of constraints (2) is the same as in the previous model. The constraint (7.3) imposes a lower bound to the quantity of intercepted flow. Also with this model it may occur that a path is intercepted by more than one facility and we do not know exactly which facility intercepts it. However, in this case for each facility the model ensures at least an unambiguous assignment, otherwise one of the facilities would be redundant.

7.6 Gain oriented problems

The models described in the previous sections adopt a performance criterion aimed to maximize the intercepted flow (*M1*) and a constraint on the amount of the total flow to intercept (*M2*), which are explicitly expressed in terms of flow values. In the following we present two models which respectively adopt a new performance criterion aimed to maximize an obtainable "gain" and a constraint on the amount of the "gain" to achieve, which are implicitly related to the flow values. The Gain Maximization Model (*M3*), finds the optimal location of a limited number of facilities for the maximization of a gain parameter (problem *P3*). The Minimum Number of Facilities for Gain Maximization Model (*M4*), finds the minimum number and the location of the facilities needed to achieve a certain amount of a gain parameter (problem *P4*). For this reason models *M3* and *M4* can be referred as "Gain Oriented". In these problems a gain coefficient a_{pi} is assigned to each node i belonging to a path p. If a node belongs to more than one path, it will be characterized by more than one coefficient and each one will be function of the flow value traversing the corresponding path. For their formulations we need to define two kinds of decisional variables, node variables y_i and path-node variables x_{pi}, which can assume the following values:

$y_i = \{0, 1\}$	*1*, if a facility is located at node i, *0* otherwise,
$x_{pi} = \{0, 1\}$	*1*, if a facility is located in the node i of the path p, *0* otherwise

It is important to note that the difference in the path variables used for models *M1* and *M2* and for *M3* and *M4* implies a great difference in the model dimension.

7.6.1 Problem *P3*: maximization of the achievable gain

Model *M3* has been proposed by Gendreau et al. [43]. Being a_{pi} the coefficient associated to each node on a path p, representing the gain obtainable if path p is intercepted by a facility located at node i, $i \in V_p$, the model can be formalized as follows:

$$\text{Maximize} \quad \sum_{p \in P} a_{pi}\ x_{pi}$$

$$\text{s.t.} \quad \sum_{i \in N} y_i = m \tag{1}$$

$$\sum_{i \in N_p} y_i \geq x_{pi} \qquad \forall p \in P, \forall i \in N_p \tag{4}$$

$$\sum_{i \in N_p} x_{pi} \leq 1 \qquad \forall p \in P \tag{5}$$

$$x_{pi} = \{0,1\} \qquad \forall p \in P, \forall i \in N_p$$

$$y_i = \{0,1\} \qquad \forall i \in N$$

The objective function expresses the maximization of the gain function. The constraint (1) is the same as in the model *M1*. The constraints (4) are consistency constraints between the two kinds of variables. Indeed, if no vertex belonging to path p contains a facility (i.e. all y_i are equal to 0), the variables $x_{pi}, \forall i \in N_p$, must be equal to *0*, i.e. its related flow is not intercepted. Otherwise, if at least a vertex of path p contains a facility (i.e. a variable y_i is equal to *1*) the variable x_{pi} could be *0* or *1*. Because of the maximization of the objective function the variable x_{pi} will be *1* and the flow on path p is intercepted. The constraints (5) impose that each path-flow can be intercepted by at most one facility located in a node belonging to the path. Therefore if a path-flow is intercepted by more facilities it will be counted only once in the objective function. Using model *M3*, contrary to what happened with model *M1*, we know exactly which is the facility intercepting a path-flow, in fact the value of variable x_{pi} returns an unambiguous correspondence between paths and facilities. Anyway, if the value of m is too high, also for problem *P3* it can happen that we could locate redundant facilities. To avoid this situation we have to substitute the constraint *(1)*, imposing that the number of facilities has to be less than or equal to m.

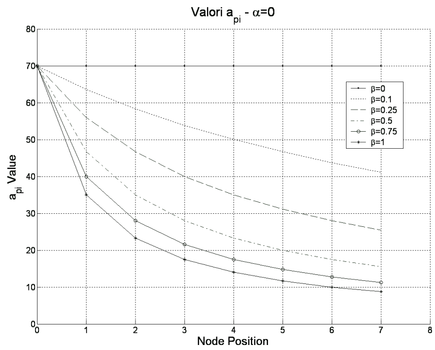
7.6.2 An empirical expression for the a_{pi} coefficients

In the original *M3* formulation the coefficients a_{pi} are assumed not decreasing along a path p, i.e. $\forall i, j \in N_p$, if i precedes j, then $a_{pi} \geq a_{pj}$. We propose here an empirical expression to compute the a_{pi} values [16]. Let d_{pod} and d_{poi} be respectively the distance between the origin and destination nodes and between the origin and a generic node i for path

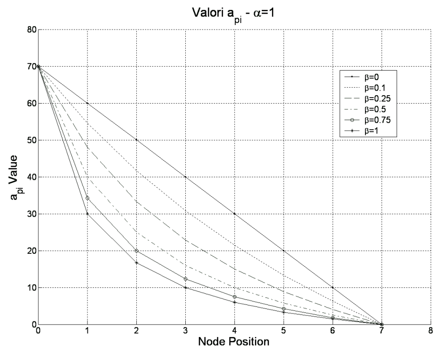
p. If P_{pi} indicates the position of a node i along a path p, assuming $P_{po} = 0$, the coefficient a_{pi} for node i on a path p can be expressed as:

$$a_{pi} = [(d_{pod} - \alpha d_{poi})/(1 + \beta(P_{pi})]f_p + R \qquad p \in P, i \in N_p \tag{6}$$

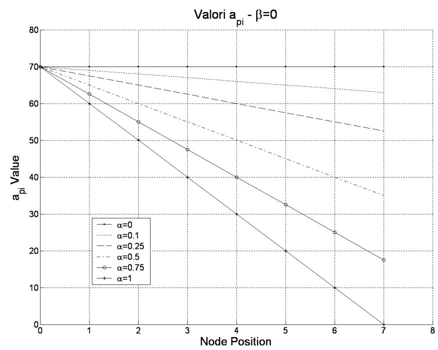
In this way the gain coefficient for a generic node i on path p is expressed as a function of its distance from the destination node $(d_{pod} - \alpha d_{poi})$, its position ($P_{pi}$) along the path and the flow value (f_p). The terms α and β are tuning parameters, assuming values between *0* and *1*, which determine the shape of a_{pi}. The addition of the constant $R \geq 0$ allows us to obtain a gain even if a flow is intercepted at its destination. Figure 7.2 shows some plots of the a_{pi} values as a function of the node position along the path, with several combinations of the tuning parameter values, in the case that the lengths of all the links are equal. We can observe that the value of a_{pi} decreases along the generic path p and the reduction rate is affected by α and β values. With these gain coefficient shapes the model returns a solution that intercepts each flow as near as possible to its origin, to which the highest gain value is associated. For this, Gendreau et al. [43] used model M3 to locate inspection stations for the interception of drunk drivers, since they are not just aimed to intercept them, but also to intercept them as soon as possible, in order to reduce the possibility of accidents. The same could be done for the interception of "bad flows" on a communication network. In fact when an hacker wants to bring an attack towards a network infrastructure, in some cases we can detect the "bad flow", but we have to block it as soon as possible, to avoid its possible success. However, it is possible to adapt the gain coefficients to the specific aim of the problem. For example, whenever it is possible to specify the node associated with the maximum gain along each path, we could hypothesize that the coefficients have a concave shape. In this way, model *M3* could be used to define a zone of the network which has to be protected by the entering flows. In fact if we make the assumption that the flows traversing a network have their origin in external nodes, whereas internal nodes are destination of the flows, we are able to identify the boundary of the area that has to be protected, by adopting an appropriate definition of the gain coefficients.



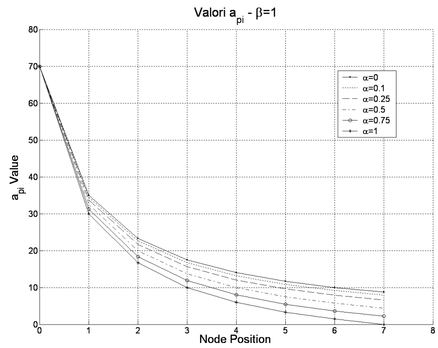
(a)



(b)



(c)



(d)

Figure 7.2: Four possible shapes of the a_{pi} coefficients.

7.6.3 Problem *P4*: minimization of the number of facilities for gain maximization

It is possible to write an inverse formulation of model *M3* that is aimed to locate the minimum number of facilities in order to achieve a predefined value of the gain function [16]. Let us define G as the global gain, achievable if each path-flow was intercepted at the node of the path having the maximum value of the gain coefficients, that is, if a_{pl} is the maximum coefficient along a path p:

$$G = \sum_{p \in P} a_{pl} \qquad l : a_{pl} = max_{i \in N_p}\{a_{pi}\}, \forall p \in P$$

Moreover, let us define $(1-\epsilon)$ as the fraction of the maximum gain G that we want to obtain $(0 \leq \epsilon < 1)$ and $G_\epsilon = (1-\epsilon)G$ the corresponding value. The model can be written as follows:

$$\text{Minimize} \sum_{i \in N} y_i$$

$$\text{s.t.} \quad \sum_{i \in N_p} y_i \geq x_{pi} \qquad \forall p \in P, \forall i \in N_p \tag{4}$$

$$\sum_{i \in N_p} x_{pi} \leq 1 \qquad \forall p \in P \tag{5}$$

$$\sum_{p \in P} a_{pi}\ x_{pi} \geq G_\epsilon \tag{7}$$

$$x_{pi} = \{0,1\} \qquad \forall p \in P, \forall i \in N_p$$

$$y_i = \{0,1\} \qquad \forall i \in N$$

The objective function expresses the minimization of the number of facilities. The set of constraints (4) and (5) are the same as in model *M3*. The constraint (7) imposes a lower bound to the value of the achievable gain to obtain.

7.7 FIFLP extensions

In this section several adaptations and integrations of the proposed models are presented. These modifications allow us to keep into account specific characteristics of a flow interception problem. Even if some of them can be used for more than one model, for sake of clearness the previous classification of the problems will be used.

Model M1: modifications and integrations
It is possible to generalize the constraint on the number of facilities to locate on the network. In fact this number is not independent by the costs. Hence being c_i ci the location cost of a single facility and C the available budget, the constraint can be modified as follows:

$$\sum_{i \in N} c_i \; y_i \leq C$$

Model M2: modifications and integrations
Several modifications can be considered for model *M2*, introducing additional constraints for one or more O/D pairs. In fact concerning the constraint related to the minimum percentage of the flow to intercept, it can be specialized for a single path or a set of paths. Being $T = \{T_i\}$ a set of disjoint subsets of the paths P and $k_{T_{i\alpha}}$ the fraction of the flow to intercept on these paths, the constraint can be modified as follows:

$$\sum_{p \in T_i} f_p \; x_p \geq T_{i\alpha} \qquad \forall T_i \in T \tag{7.4}$$

Moreover in order to intercept at least R paths of a subset $T_i \subseteq P$ the model can be integrated with the following constraint:

$$\sum_{p \in T_i} x_p \geq R \qquad \forall T_i \in T \tag{7.5}$$

Finally to guarantee that at least h facilities intercept a path p, the following model can be added:

$$\sum_{i \in N_p} y_i \geq h \qquad \forall p \in T_i \tag{7.6}$$

Model M3: modifications and integrations
Two possible modifications can be considered for model *M3*, concerning the maximum interception capacity of the facilities and the multiple interception of the flows. Obviously also for model *M3* it is possible to modify the constraint on the number of facilities as it has been shown for model *M1* with the budget constraint.

In the previous formulation each facility could intercept all the flows traversing it. Anyway it can happen that a facility has a limited interception capacity, which cannot be ignored. Therefore being L_i the set of the paths p that traverse the node i and q_i the capacity of the i^{th} facility, the following constraint can be added:

$$\sum_{p \in L_i} f_p \; x_{pi} \leq q_i \; y_i \qquad \forall i \in N \tag{7.7}$$

The introduction of this constraint allows to introduce the possibility of performing sampling operations or share the amount of the intercepted flows. To this aim the model *M3* can also be modified with the relaxation of the binary constraints for the variables x_{pi}.

A second modification for model *M3* consists in the generalization on the constraint related to the multiple counting of the flows. Therefore being $T = \{T_i\}$ a set of disjoint subsets of P and h_i the maximum number of times that each subset T_i can be intercepted, the constraint can be modified as follows:

$$\sum_{i \in N_p} x_{pi} \leq h_i \qquad \forall p \in T_i \tag{7.8}$$

Since the maximization of the objective function in model *M3* more than one variable for each path could assume value 1 in the solution set.

Model M4: modifications and integrations The modifications and integrations for model *M4* can be easily derived by the previous sections. In fact in this model constraints already presented for the model *M2* and for model *M3* can be easily adapted and introduced. In particular being $T = \{T_i\}$ a set of disjoint subset of P and $G_{T_{i\alpha}}$ the fraction of the maximum achievable gain for this subset, then the following constraint can be added:

$$\sum_{p \in T_i} a_{pi} \; x_{pi} \geq G_{T_{i\alpha}} \qquad \forall T_i \in T \tag{7.9}$$

Moreover, concerning the multiple interception of the flows, the only difference with the constraint proposed for model *M3* is in the fact that value h represent the exact number that each path or a set of paths has to be intercepted. Therefore being $T = \{T_i\}$ the set of disjoint subsets of P to intercept, the constraint can be modified as follows:

$$\sum_{i \in V_p} x_{pi} = h_i \qquad \forall p \in T_i \tag{7.10}$$

7.8 Mobile facility location problem

A facility location problem is a problem of spatial allocation. In these problems, as it has been shown in previous sections, the aim is to optimise an objective function, or a set of objective functions, under the definition of several restrictions, referred as constraints.

In literature the location problem have been quite always considered for networks, where the cost and the traveling time for each O/D pair were known and constant. However, in many context and in particular in the urban areas, the parameters characterizing the state of the network can vary with the time or could be very uncertain. Therefore we can say that the parameters of the network can vary with the traffic condition and can change from a time slot to another.

For this reason it is interesting to treat the case of mobile facilities, which can change their location during a time horizon depending on the status of the network and in particular, in our case, the varying parameter is the flow value for the O/D pairs. This means that the problem is treated as multi-stage problem.

For each of the stage of the network the optimal location of the mobile facilities has to be determined. For the models that are going to be presented, the parameter changing from a status to another, is the flow value for the O/D pairs. Therefore, the status of the network change for at least a flow value from a stage to another.

The models for the location of mobile facilities can be derived from the model *M1* and *M2*. In particular two approaches are considered:

1. iterative approach

2. dynamic approach

The iterative approach solves at each stage the model *M1* and then an assignment problem is solved to find the optimal relocation of the facilities, i.e. the relocation minimizing the costs. For what concerns instead the dynamic location, we can generalize the models *M1* and *M2* to the multi-stage case.

7.8.1 Iterative approach

We will refer, as for the previous models, to a network *G(N,A)*, where $N = \{i\}$ is the set of nodes and $A = \{l\}$ is the set of links. Moreover we will define $P = \{p\}$ as the set of origin-destination paths selected on the network, N_p as the set of nodes belonging to a path p, $F = \{f_p\}$ as the set of flows on the paths belonging to P and m as the number of facilities to locate on the network at each stage. At constant time slots the network passes from a status to another. The transition among the different status of the network is given by a T' transition matrix, where the generic element t_{rs} represents the probability of a transition from status r to status s of the network.

The iterative approach takes his name from the fact that to find the optimal location of the m mobile facilities, the model $M1$ is solved iteratively for each status of the network. Then an assignment problem is solved to determine the association between old and new location of the facilities, in order to minimize the relocation costs. Here just the assignment problem to solve is presented.

If i is the index related to the set of the open facilities at current stage, and j the location of the facilities at the successive stage, then a variable w_{ij} is defined, which assumes the following values:

$w_{ij} = \{0, 1\}$ *1*, if a facility located at node i is relocated at node j, *0* otherwise

Being c_{ij} the relocation cost in terms of traveling time to move a facility from the node i to j, the problem is the following:

$$\text{Minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij}\ w_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} x_{ij} = 1 \qquad \forall i \in (1, ..m) \tag{7.11}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \in (1, ..m) \tag{7.12}$$

$$x_p = \{0, 1\} \qquad \forall p \in P$$

The objective function expresses the minimization of the relocation costs. Constraints (7.11) impose that each mobile facility can be assigned to just one new location, whereas the constraints (7.12) impose that each new location can be location of just one facility.

7.8.2 Dynamic approach

The second approach for the mobile facility location problems is the dynamic approach. Two models will be proposed, the first derived from the model *M1* and the second from model *M2*.

The first model, referred as dynamic *M1*, *DM1*, allows to determine the set of locations on the network for the m facilities at each stage, maximizing the intercepted flow, without double counting operations and minimizing the relocation costs.

In the formulation of this problem two variables have to be defined:

$x_{ijk} = \{0, 1\}$	*1*, if at status i, the facility j, is located at node k, *0* otherwise
$y_{ip} = \{0, 1\}$	*1*, if a located facility at stage i is on the path p, *0*, otherwise

Moreover a concave and not decreasing objective function is defined, which is given by two components A and B:

$$A = \sum_{i=1}^{S} \sum_{p=1}^{P} \ F(-f_{ip}) \ y_{ip}$$

$$B = \sum_{i=1}^{S} \sum_{l=1|l \neq i}^{S} p_{il} \left\{ \sum_{j=1}^{m} \sum_{k=1}^{n} \sum_{q=1|q \neq k}^{n} \ xijk * \ x_{ljq} \ f[d(k, q)] \right\}$$

The factor F is a tuning factor, which is used to homogenize the two cost components. In fact in A we have the intercepted flow at each stage and in B we have the relocation costs in terms of traveling time. The minus sign before the f_{ip} is used because our aim is to maximize the intercepted flow, but the whole objective function has to be minimized.

Therefore the model *DM1* is the following:

$$\text{Minimize} \quad A + B$$

$$\text{s.t.} \quad \sum_{j=1}^{m} x_{ijk} \leq 1 \forall i \in S, \forall k \in N \tag{7.13}$$

$$\sum_{k \in N} x_{ijk} = 1 \forall i \in S, \forall j \in (1, .., m) \tag{7.14}$$

$$y_{ip} \leq \sum_{k \in P} \sum_{j=1}^{m} x_{ijk} \forall p \in P, \forall i \in S \tag{7.15}$$

$$x_{ijk} = \{0, 1\} \qquad \forall i \in S, \forall j \in (1, .., m), \forall k \in N$$

$$y_{pi} = \{0, 1\} \qquad \forall p \in P, \forall i \in S$$

Constraints (7.13) and (7.14) impose that at each stage at each node it can be located no more than one facility and that at each node just one facility can be located. Constraints (7.15) finally impose that y_{ip} is equal to *0* if all the x_{ijk} for a path k are equal to 0, i.e. no facilities are located on the path at stage i. On the other side, if there is at least a facility located on the path at stage i, then at least a variable x_{ijk} is equal to 1 and for this reason the variable y_{ip} can assume value 1 or 0. But, since in the formulation the A component of the objective function has to be minimized, then the y_{ip} will be forced to assume value *1*. Moreover since the variable y_{ip} can assume just value 0 or 1, then no double counting of the flows is possible.

The last model that we are going to present is the model derived from the *M2* model, but in the dynamic case, referred as *MD2*. This model tries to determine the number of mobile facilities that we need to locate at each stage in order to intercept at each stage a predefined percentage of the flow traversing the network, minimizing the relocation costs. This model uses the same set of variables used for model *MD1* and also in this case the objective function is given by two components, differing for the expression related to the A component:

$$A = \sum_{i \in S} \sum_{j \in (1,..,m)} \sum_{k \in N} \; c \; x_{ijk}$$
$$B = \sum_{i=1}^{S} \sum_{l=1|l \neq i}^{S} \; p_{il} \left\{ \sum_{j=1}^{m} \sum_{k \in N} \sum_{q \in N_{q \neq k}} \; xijk * \; x_{ljq} \; f[d(k,q)] \right\}$$

The A component is related to the location cost at each stage, whereas the B component is related to the relocation cost. The facto c in the A component represents the location cost for the facility, and it has to be paid each time a facility is used.

The problem can be formulated as follows:

$$\text{Minimize} \quad A + B$$

$$\text{s.t.} \quad \sum_{j=1}^{m} x_{ijk} \leq 1 \forall i \in S, \forall k \in N \tag{7.16}$$

$$\sum_{k \in N} x_{ijk} = 1 \forall i \in S, \forall j \in (1,..,m) \tag{7.17}$$

$$\sum_{j \in (1,..m)} \sum_{k \in N} x_{ijk} \geq 1 \forall i \in S \tag{7.18}$$

$$\sum_{i \in S} \sum_{p \in P} f_{ip}\, y_{ip} \geq C^*_\alpha \tag{7.19}$$

$$y_{ip} \leq \sum_{k \in P} \sum_{j=1}^{m} x_{ijk} \forall p \in P, \forall i \in S \tag{7.20}$$

$$\sum_{k \in N} x_{ijk} = \sum_{k \in P} x_{rjk} \forall i, r \in S_{r \neq i}, \forall j \in (1, .., m) \tag{7.21}$$

$$x_{ijk} = \{0, 1\} \qquad \forall i \in S, \forall j \in (1, .., m), \forall k \in N$$

$$y_{pi} = \{0, 1\} \qquad \forall p \in P, \forall i \in S$$

Constraints (7.16) and (7.17) impose respectively that at each stage at each node no more that one facility can be located and that at each node one facility can be located. Constraints (7.18) impose that at each stage at least a facility has to be located. Constraints (7.19) impose that at least a percentage C^*_α has to be intercepted. Constraints (7.20) impose that y_{ip} is equal to 0 if all the x_{ijk} for the path k are equal to *0*, i.e. no facilities are located on the path p. On the other side if at least a facility is located on the path at stage i, then at least a variable x_{ijk} assumes value *1* and for this reason the y_{ip} can be equal to 1 or 0. But since the problem tries to minimize the objective function then the y_{ip} will be forced to assume value 1 and no double counting is allowed. Finally the constraints (7.21) impose that at each stage the number of facilities to locate has to be the same and that if a facility is used in a stage it will be also used at the successive one.

The dynamic approach, differently from the iterative approach, is applied just once. In fact it returns all the solutions for all the stages at the same moment. The characteristic of this model is that it is able to better manage the relocation from a stage to another. In fact in the iterative approach the optimal locations at each stage are determined in order to maximize the intercepted flow and then it is evaluated the relocation cost. Whereas in the dynamic approach the relocation cost is taken into account on the whole time horizon. For this reason it is easy to say that we expect to find better solution from the point of view of the intercepted flow, with the usage of the iterative approach, whereas we expect to find better solution from the point of view of the relocation costs, using the dynamic approach.

Chapter 8

Heuristic approaches for FIFLP

In the previous chapter models for the flow interception facility location problem have been presented. These models are related to NP-hard problems. In fact the solutions of these problems are represented by all the possible combinations of k facilities on n nodes, with k varying in the range $[0, n]$. This means that the number of possible solutions corresponds to $\sum_{k=0}^{n} \binom{n}{k} = 2^n$ and therefore the computation times will exponentially increase with the number of nodes. For this reason it is necessary to use heuristic approaches to solve these problems on large instances.

8.1 Two greedy heuristics for problem *P1*

For large size instances problem *P1* can be solved by the greedy heuristic (*H1*) proposed in [13] in terms of node location problem (of discretionary service facilities) and reformulated in [109] in terms of link location problem (of road traffic sensors). With the previous notation $\boldsymbol{B}(k)$ is the path-node incidence matrix at stage k, $\boldsymbol{b}_i(k)$ the vector corresponding to the i^{th} column of $\boldsymbol{B}(k)$, $\boldsymbol{f}$ the vector of the flow values related to the p paths ($p \in P$) and S the node solution set, the algorithm can be structured as in the following:

Step 0. Set $k = 0$, $S = \{\oslash\}$ and let $\boldsymbol{B}(k)$ be the corresponding coverage matrix.

Step 1. Compute the flow interceptable by each node i at stage k:
$f_i(k) = \boldsymbol{f} \cdot \boldsymbol{b}_i(k)$, $\forall i \in N \backslash S$.

Step 2. Find the node i^*: $f'_{i^*}(k) = max_{i \in N} f_i(k)$ and locate a facility in the node i^*. $S = S \cup \{i^*\}$. If more than one, choose the node with the lowest index or the node belonging to the greatest number of paths.

Step 3. Update ***B(k)*** elements to generate ***B(k+1)***.
$b_{pi|i \in S}(k+1) = 0 \qquad \forall p \in P$
$b_{pi}(k+1) = 0 \qquad \forall i \in V, \forall p \in N | b_{pi|i \in S}(k) = 1$

Step 4. If $|S| = m$, then STOP. Else set $k = k+1$ and return to ***Step*** 1.

Heuristic *H1* returns the location of the m facilities intercepting a certain amount of flow. It can be improved in order to obtain better solutions in terms of number and location of facilities [16]. The modification differs from the basic heuristic for the definition of two flow parameters for each node i:

- $f'_i(k)$ is the flow interceptable by node i at iteration k, as in the basic heuristic,
- $f''_i(k)$, which will be referred to as potential flow, is the amount of flow already intercepted by other selected nodes at iteration k.

Using the parameters defined in 7.5, this algorithm, referred as *Improved H1* is structured as follows:

Step 0. Set $k = 0$, $S = \{\oslash\}$ and let ***B(k)*** be the corresponding coverage matrix.

Step 1. $\forall i \in N \backslash S$, compute $[f'_i(k), f''_i(k)]$.
$f'_i(k) = \boldsymbol{f} \cdot \boldsymbol{b}_i(k) \qquad f''_i(k) = f'_i(0) - f'_i(k)$

Step 2. Sort the nodes in decreasing lexicographic order with respect to the couple $[f'_i(k), f''_i(k)]$, locate a facility in the first node and add it, referred as i^*, to the set S ($S = S \cup \{i^*\}$).

Step 3. Update ***B(k)*** elements to generate ***B(k+1)***.
$b_{pi|i \in S}(k+1) = 0 \qquad \forall p \in P$
$b_{pi}(k+1) = 0 \qquad \forall i \in N, \forall p \in P | b_{pi|i \in S}(k) = 1$

Step 4. If $|S| = m$ then go to the ***Step 5***.
Else set $k = k + 1$ and return to ***Step 1***.

Step 5. Set $S = T$.
$\forall l \in T$, *if* $\sum_{j \in S \setminus \{l\}} b_{pj} \geq 1, \forall p \in P | b_{pl}(0) = 1$, then $S = S \setminus l$, otherwise S remains unchanged.

Step 6. If $|S| = m$, then STOP. Else set $k = k + 1$ and return to ***Step*** 1.

The use of the parameter $f_i''(k)$ generates solutions where several flows can encounter more than one facility along their path. It can thus occur that some facilities become redundant. In this case, by means of step 5, we can eliminate the redundant facility from the solution set and relocate it in another node, without variation of the m value and with a possible increase of the amount of intercepted flow. In solving the problem by these two heuristics, it can happen that the number of facilities is too high. In this case, as explained for the model $M1$, we could add a new stop criterion respectively in the Step 4 of *H1* and in the Step 6 of *Improved H1*, which imposes to stop if all the elements of the matrix ***B(k)*** are equal to *0*, i.e. all the flows traversing the network are already intercepted.

8.2 An ascent heuristic for *P2*

A simple heuristic approach (*H2*) for problem *P2* could be derived from that used for problem *P1*, opportunely modifying the stopping criterion [13]. In fact, following the same steps of *H1*, the heuristic stops when the intercepted flow is equal or greater to the prefixed value $K_\epsilon(0 \leq \epsilon < 1)$. Berman et al. [13] proposed also an ascent heuristic which returns the optimal solution for problem *P2* by solving model *M1* or the relaxed model of *M1* (obtained by replacing the integrality constraints with $0 \leq x_p \leq 1$ and $0 \leq y_i < 1$). Here we present a slightly modified version of it to explicitly compute the intercepted flow at every step. If *F(k)* is the flow intercepted at stage k and *RM1* is the relaxed model of *M1*, the heuristic proceeds as follows:

Step 0. Set $k = 0$ and let ***B(k)*** be the corresponding coverage matrix.
$F(0) = 0$. $S = \{\emptyset\}$.

Step 1. Compute $f_i(k) = \boldsymbol{f} \cdot \boldsymbol{b}_i(k)$, $\forall i \in N \backslash S$.

Step 2. Find node $i^* : f_i(k) = max_{i \in N} f_i(k)$ and locate a facility in node i^*.
$F(k+1) = F(k) + f_i(k)$

Step 3. Update ***B(k)*** elements to generate ***B(k+1)***.
$b_{pi|i \in S}(k+1) = 0 \qquad \forall p \in P$
$b_{pi}(k+1) = 0 \qquad \forall i \in N, \forall p \in P | b_{pi|i \in S}(k) = 1$

Step 4. If $F(k+1) \geq K_\epsilon$, then set $m = k$ and go to *Step 5*.
Otherwise set $k = k + 1$ and return to *Step 1*.

Step 5. Set $m = m - 1$ and solve the model *RM1* with m facilities. If the objective function value of *RM1*, $z^*(RM1)$, is less than or equal to K_ϵ, go to *Step 6*, otherwise solve problem *P1* to optimality. If the solution of model *M1* returns a value of objective function greater than K_ϵ, $z^*(M1)$, then repeat *Step 5*, otherwise go to *Step 7*.

Step 6. Set $m = k$. $F(k) = z^*(RM1)$ and STOP.

Step 7. Set $m = k$. $F(k) = z^*(M1)$ and STOP.

The reason for which *H2* returns the optimal solution for *P2* can be explained as follows. The m value of *step 4* is an upper bound to the optimal value of the objective function of *M2*. *RM1*, solved fixing $m = m - 1$, provides an upper bound to model *M1*. Therefore, if this upper bound is less than or equal to K_ϵ, the exact solution of *M1* cannot provide a feasible solution with optimal objective value larger than K_ϵ. Once the optimal solution value m, required to intercept a flow greater or equal to K_ϵ, is obtained, *M1* can be solved to provide a set of locations possibly with a better total flow [13].

Finally, we observed that this approach can also be used in case we want to intercept the entire flow traversing the network ($\epsilon = 0$). In this case the objective function value at *STEP 5*, obtained by solving *RM1* and *M1* with $m = m - 1$, must be strictly less and not less than or equal to K_ϵ, otherwise we have no reduction in the number of facilities. It is important to underline that even if this heuristic allows us to find the exact solution for problem *P2*, it could be not convenient to use. In fact, it is clear that we could obtain the optimal solution just solving the relaxed model *RM1*, but it could also happen that we have to solve model *M1* many times and this should imply a significant increase in the computation time.

8.3 Heuristic for multiple FIFLP

The greedy heuristic for problem P2 can be opportunely modified in order to intercept each path more than ones. To this aim two arrays of dimension $|P|$ are defined: $\zeta(k)$ and $\gamma(k)$. The generic element of the first array, $\zeta_p(k)$, indicates the number of times that each path p has still to be intercepted at stage k, whereas the generic element of the second array, $\gamma_p(k)$ the number of times that each path p has been already intercepted at stage k.

The algorithm can be formalized as follows:

Step 0. Set $k = 0$, $\gamma_p(k) = 0\ \forall p \in P$, $S = \{\oslash\}$ and let $\boldsymbol{B}(k)$ be the corresponding coverage matrix. Moreover set $\zeta_p(k)$ equal to the number of times that each path has to be intercepted.

Step 1. Compute the flow interceptable by each node i at stage k:
$f_i(k) = \boldsymbol{f} \cdot \boldsymbol{b}_i(k)$, $\forall i \in N \backslash S$.

Step 2. Find the node i^*: $f'_{i^*}(k) = max_{i \in V} f_i(k)$ and locate a facility in the node i^*. $S = S \cup \{i^*\}$. If more than one, choose the node with the lowest index or the node belonging to the greatest number of paths.

Step 3. Update ***B(k)***, $\zeta(k)$ and $\gamma(k)$, elements to generate ***B(k+1)***, $\zeta(k+1)$ and $\gamma(k+1)$.
$b_{pi|i \in S}(k+1) = 0 \qquad \forall p \in P$
$b_{pi}(k+1) = 0 \qquad \forall i \in N, \forall p \in P | b_{pi|i \in S}(k) = 1$ and $\zeta_p(k+1) = 0$
$\gamma_p(k+1) = \gamma_p(k) - 1 \qquad \forall p \in P | b_{pi|i \in S}(k) = 1$
$\zeta_p(k+1) = \zeta_p(k) + 1 \qquad \forall p \in P | b_{pi|i \in S}(k) = 1$

Step 4. If $b_{pi}(k+1) = 0, \forall i \in N, \forall p \in P$, then go to ***Step*** 5. Else set $k = k + 1$ and return to ***Step*** 2.

Step 5. If $\zeta_p(k+1) = 0, \forall p \in P$, then set $m = k$ and STOP. Else the problem cannot be solved with the imposed values for $\zeta(k)$ and $\gamma(k)$.

The heuristic returns the number and the location of the facilities required for the multiple interception of the flows and it terminates when the two arrays $\zeta(k)$ and $\gamma(k)$ are equal. It is important to note that the problem will have no feasible solution if the number times that each path has to be intercepted is higher than the number of nodes composing the path for an O/D pair.

8.4 A greedy heuristic for *P3*

A greedy heuristic (*H3*) for this problem, proposed in [43], determines a feasible solution by a sequential selection of the nodes providing the maximum increase of the objective function. S_k being the set of solution nodes at stage k and $z(S)$ being the related objective function value, the heuristic proceeds as follows:

Step 0. Set $k = 0$ and $S_0 = \{\oslash\}$.

Step 1. Set $k = 1$. $S_k = S_{k-1} \cup \{i^*\}$, where i^* is the node providing the maximum increase of the objective function:

$$max_{i \in N \setminus S_{k-1}}[z(S_{k-1} \cup \{i\}) - z(S_{k-1})]$$

Step 2. If $|S| = m$ Stop, otherwise set $k = k + 1$ and repeat *Step 1.*

Heuristic *H3*, as the others previously described for problem *P1*, is a "greedy heuristic", in fact at every iteration it introduces the node which is the most profitable in the partial solution set, without taking into account the global solution. Therefore, the solutions provided by this method can be improved through the use of local search methods. In the following we describe two heuristics, a local ascent search and a tabu search heuristic, proposed by Gendreau et al. [43] for the improvement of the solution provided by *H3*. Also this heuristic can be modified in order to avoid to locate redundant facilities, just imposing to stop if no vertex $i \in N \setminus S_{k-1}$ provides an increase of the objective function which is strictly greater than zero.

8.5 An ascent search heuristic for *P3*

This heuristic, starting from a first feasible solution, determined with greedy heuristic *H3* or randomly generated, tries to improve it by substituting a node belonging to the solution with a node not belonging to

it. For this operation, at any iteration we characterize the solution by the definition of the subset of nodes composing it ($S \subseteq N$), and by the subset of its neighborhood $N(S)$, which is constituted by all solutions achievable by performing an interchange, i.e. a substitution of a solution node i, with all nodes *i'* of the set $N \backslash S$ (this operation is referred to as 'move'). We have several rules for the choice of nodes i and *i'*.
Five rules for the choice of vertex i are considered:

Rule 1. The node associated to the minimum value of the objective function:

$$min_{i \in S}\{\sum_{p \in P} a_{pi}\}.$$

Rule 2. The node belonging to the minimum number of paths.

Rule 3. A node randomly chosen in the solution set S.

Rule 4. The node that gives the minimum increase in the value of the objective function:

$$min_{i \in S}\{z(S) - z(S \backslash \{i\})\}$$

Rule 5. All nodes i of S are chosen for the interchange operation and the one that returns the most profitable interchange is selected.

A possible rule for the choice of node *i'* consists of selecting the node that returns the maximum increase of objective function, that is:

$$max_{i' \in N \backslash S}\{z(S \backslash \{i\}) \cup \{i'\}\}$$

If S^* is the current solution set, the ascent search heuristic (ASH) for problem *P3* can be formalized as follows:

Step 0. Determine an initial solution S. Set $S^* = S$ and $k = 1$.

Step 1. According to one of the five rules just described, choose node i and make the interchange of node i with all nodes $i \in N \backslash S$. Choose between all possible interchanges, the one associated with the maximum increase of objective function.

Step 2. If $z(S') \leq z(S^*)$, the algorithm stops, otherwise set $S^* = S'$ and return to *Step 1.*

8.6 A tabu search heuristic for *P3*

The ascent search returns a solution corresponding to a local maximum for problem *P3* and so we can try to improve it by a tabu search heuristic (*TSH*). This method, starting from a solution S determined with *H3* or randomly generated, follows the same structure of the *ASH* just described, but *TSH* can return a worse solution from one iteration to the next. This method foresees a mechanism to avoid the repetition of a solution, by declaring the ones which possess some attributes as forbidden. It uses the same rules previously defined for the choice of node i and *i'*. The basic differences are:

1. deteriorating solutions can be accepted,

2. the repetition of the same solution can be avoided by declaring it as tabu,

3. the stopping rule is different. It is defined as the number of maximum iterations to implement, or the maximum pre-defined number of iterations, (δ^*), without solution improvement.

With the given notations, if δ is the current number of iterations without a solution improvement and *S'* is the solution set after a move, *TSH* algorithm can be described as follows:

Step 0. Determine an initial solution S. Set $S^* = S$, $k = 1$ and $\delta = 0$. No (i, i') pair is declared as tabu.

Step 1. Determine the solution corresponding to the best (i, i') move, S'. The following cases can occur:

- if (i, i') is not tabu and $z(S') > z(S^*)$, set $S^* = S'$ and $\delta = 0$,
- if (i, i') is not tabu and $z(S') \leq z(S^*)$, set $S^* = S'$ and $\delta = \delta + 1$,
- if (i, i') is tabu and $z(S') > z(S^*)$, set $S^* = S'$ and $\delta = 0$,
- if (i, i') is tabu and $z(S') \leq z(S^*)$, choose the best non tabu move belonging to $I(S)$. Let us indicate it as (I, I') and the corresponding solution , S''. Therefore:
  If $z(S'') > z(S^*)$, set $S^* = S''$ and $\delta = 0$
  otherwise set $S^* = S'$ and $\delta = \delta + 1$.

If $\delta = \delta^*$, STOP. Otherwise set $k = k + 1$, declare (i, i') and (i', i) or (I, I') and (I', I) as tabu moves for a pre-defined number of iterations and repeat *Step 1*.

The use of *TSH* could find a local maximum point better than the one determined with *ASH* and in the best case finds the global optimum. It is important to note that both ascent and tabu search heuristics, can be easily adapted in order to use them to find a better solution for problem *P1*. In fact problem *P1* can be considered as a special case of problem *P3*, where the gain coefficients values a_{pi} are equal to flow values along path p. Therefore, in order to use these methods for gain maximization, we just need to replace the objective function of *P3* in the steps just described with the one in problem *P1* and the a_{pi} values with flow values f_p.

8.7 A greedy heuristic for *P4*

For this problem we propose here a greedy heuristic (*H4*). Let us define ***A**(k)* as the gain matrix at stage k, where the generic element is equal to the gain coefficients a_{pi}, for node i on path p. Let also $\boldsymbol{a}_i(k)$ be the vector corresponding to the i^{th} column of ***A**(k)*. Being $G(k)$ the gain achieved at stage k and using the notation previously introduced, the algorithm can be formalized as follows [16]:

Step 0. Set $k = 0$. $S = \{\oslash\}$. $G(0) = 0$.

Step 1. Compute $a_i(k) = \sum_{p \in P} a_{pi}(k)$, $\forall i \in N \backslash S$.

Step 2. Find the node $i^* : a_{i^*}(k) = max_{i \in N}\{a_i(k)\}$. Locate a facility in i^* (if more than one, choose the node with the lowest index, or belonging to the greatest number of paths, or corresponding to the higher a_{pi}).
$G(k+1) = G(k) + a_{i^*}(k)$

Step 3. Compute: $\forall p \in P | i \in N_p, m_{pi}(k) = a_{pi}(k) - a_{pi|i \in S}(k)$.

Step 4. Update the generic element a_{pi} and generate the matrix $G(k+1)$.
$\forall p \in p | i \in N_p$, if $m_{pi}(k) > 0$, then $a_{pi}(k+1) = m_{pi}(k)$
$\forall p \in p | i \in N_p$, if $m_{pi}(k) < 0$, then $a_{pi}(k+1) = 0$
$\forall p \in p | i \notin N_p$, $a_{pi}(k+1) = a_{pi}(k)$

Step 5. If $G(k+1) \geq G_\epsilon$, then STOP, otherwise set $k = k+1$ and return to *Step 1*.

This heuristic can be easily adapted to find a solution for problem *P3*, just changing the stop criterion as follows:

Step 5bis. If $k = m$ or if $a_{pi} = 0$, $\forall i \in N$ and $\forall p \in P$, then STOP, otherwise set $k = k+1$ and return to *Step 1*.

This modified heuristic will be referred to in the following as *Improved H3* and the stopping criterion on the values of the gain coefficients allows us to avoid to locate redundant facilities.

Chapter 9

Computational results for FIFLP

In this chapter the mathematical models and heuristic methods of the previous chapters have been experienced on test networks of varying dimension and topology (mesh and random), comparing the obtained results in terms of quality of solution and computation times. We experienced models and methods varying some settings and characteristics of the problems under investigation (for example range of the flow values, number of paths and facilities) in order to verify the effect of these parameters. Original graphical representation of the obtained results are provided.

9.1 Experimental tests on grid and random networks

The experimental results have been obtained by solving the four problems described in the previous chapter, to the optimum, by using the Xpress-MP solver, and to a feasible solution, by using the described heuristics, coded in *C* language. All the instances were run on an Intel(R) Pentium(R) 4 computer. In our tests we used synthetic networks of increasing dimensions with random topology (50, 100, 150 and 200 nodes) and grid topology (49, 100, 144, 196 nodes). We note that for all problems, each flow is assigned to the shortest path, assuming random and symmetric link cost. The values of a_{pi} coefficients for problem *P3* and *P4* are assumed to be decreasing along the path and are computed by expression *(8)* of section 7.6.2, with $\alpha = 1$, $\beta = 0$ and $R = 1$.

Network topology and dimension	Grid: 49, 100, 144, 196 nodes Random: 50, 100, 150, 200 nodes
Range flow values	F_p: 1-10; 1-30; 1-50
Device number	$\|N\| = 50$. m = 1,. . .,$\|N\|$; step size: 1. $\|N\| = 100$. m = 1,. . .,$\|N\|$; step size: 5. $\|N\| = 150$. m = 1,. . .,$\|N\|$; step size: 10. $\|N\| = 200$. m = 1,. . .,$\|N\|$; step size: 10.
Number of instances	I = 50
Max number of iterations with no improvement	$\delta = 5$. $\delta = 20$.
Percent of flow value	$0 \leq \epsilon < 1$. Step size: 0.01. $K_\epsilon = (1 - \epsilon) \cdot K$
Percent of gain value	$0 \leq \epsilon < 1$. Step size: 0.01. $G_\epsilon = (1 - \epsilon) \cdot G$
Tuning parameters	$\alpha = 1$; $\beta = 0$; $R = 1$.
Number of paths	10%, 20%, 40%, 80% of the $o - d$ pairs number.

Table 9.1: Test Network and Parameter Setting.

9.2 Graphical representation of the experimental results on a small network

The synthetic network under investigation is a random network with *50* nodes, which are all origin/destination nodes. We assume that each node is origin of at least a path. We randomly generated *250* paths, corresponding to about *10%* of the o-d pairs number. Each point of the following figures is determined as an average of *50* instances, differing for the flow values traversing the predefined paths. The four defined problems have been solved using the same instances.

9.2.1 Intercepted flows vs. number of facilities for *P1* and *P2*

Figure (9.1) shows the results obtained by solving problem *P1* with model *M1* and related heuristics (*H1*, *Improved H1*, *ASH* and *TSH* using rule 5, i.e. the rule providing the best results for these two methods), plotting the percent of the intercepted flow as a function of the number of located devices. We can easily observe that the distance between the five trends is not really significant; however, between *3* to *16* devices, a small distance can be perceived. Indeed in all cases we intercept more than *90%* of the total flow locating *9* devices (i.e. *18%* of possible locations). Moreover, we intercept the entire flow with *16* devices (i.e. *32%*) using *M1*, *19* (i.e. *37%*) using *ASH* and *TSH* and *21* (i.e. *43%*) using *H1* or the improvement of *H1*. Figures (9.2) and (9.3) show the

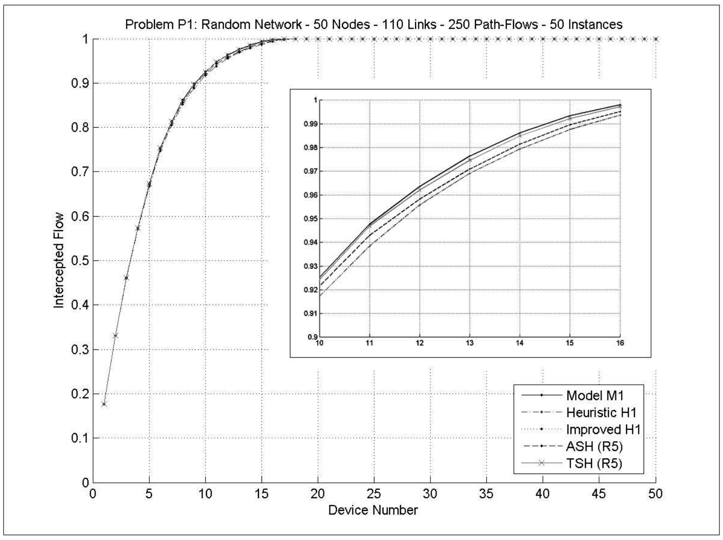


Figure 9.1: Problem P1: Model M1 solutions Vs Heuristic method solutions.

results achieved with the five different rules described in section 8.5 and 8.6, respectively for *ASH* and *TSH*. We can observe that in both cases the worst results are related to rule 1 and 2, whose maximum difference to the optimum is *0.0183* for *ASH* and *0.0178* for *TSH*. Instead the best results obtained with rule 5 are characterized by a maximum difference to the optimum equal to *0.003* for both methods. It is also important to note that rule 3 (random selection of the node to substitute) returns better results than the other three rules.

Figure (9.4) and (9.5) show the results obtained by solving problem *P2* with *M2* and *H2*, plotting on the X-axis the number of devices as a function of the percentage of the intercepted flow ($(1 - \epsilon)$, $0 \leq \epsilon < 1$ step size *0.01*). In figure (9.4) we show the results of the model and the heuristic for a single instance. It is important to note that the obtained trend, both for *M2* and *H2*, can be assimilated to a step function, since

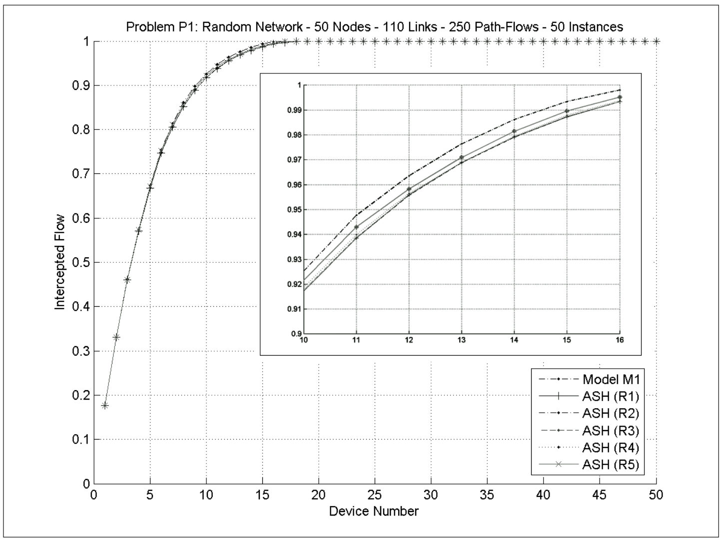


Figure 9.2: Problem P1: Model M1 solutions Vs ASH solutions (five rules).

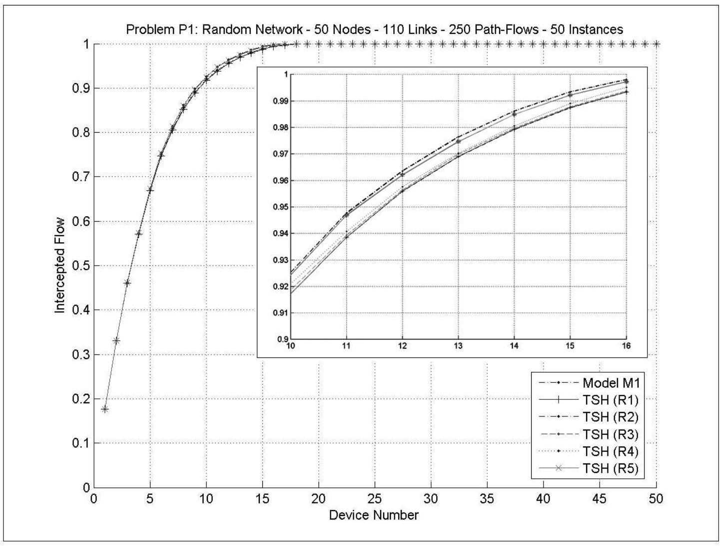


Figure 9.3: Problem P1: Model M1 solutions Vs TSH solutions (five rules).

we always have an unambiguous correspondence between a fixed value of the intercepted flow and the number of required devices. It is also important to observe that the results obtained solving model *M2* and the ones obtained solving the heuristic *H2* are exactly the same. This is what we expected, since, as explained in section 8.2, this heuristic finds the optimal solution for problem *P2*. In figure (9.5) we show the results obtained solving *P2* by *M2* and *H2* for *50* instances differing for the flow patterns. We can see that for a fixed percent of the intercepted flow, we can have several values of the number of required devices. Obviously this is due to the fact that solving instances differing for the flow patterns, the number of devices to locate intercepting the same amount of flow could be different.

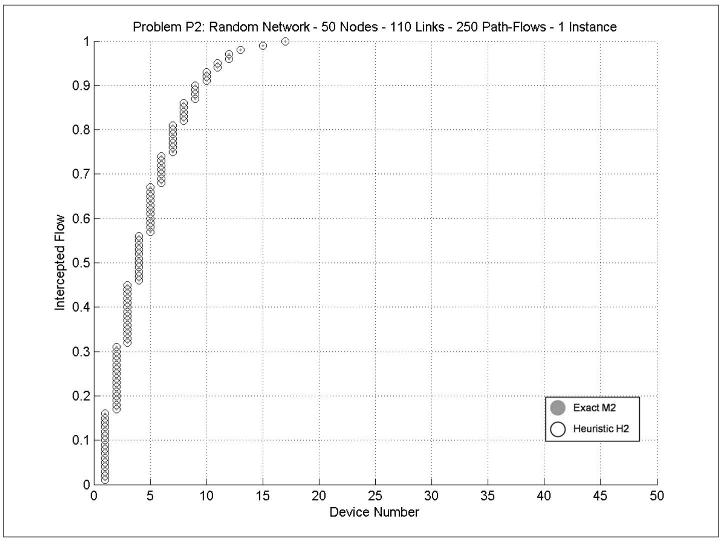


Figure 9.4: Problem P2: Model M2 solution vs Heuristic H2 solutions over a single instance.

9.2.2 Achieved gain vs. number of facilities for problem *P3* and *P4*

Figure (9.6) shows the results obtained solving problem *P3* with model *M3* and related heuristics (*H3*, *ASH* and *TSH* using rule 5), plotting the percentage of the achieved gain as a function of the number of located devices. As seen for problem *P1*, in this case we can observe that the distance between the five trends is not really significant; how-

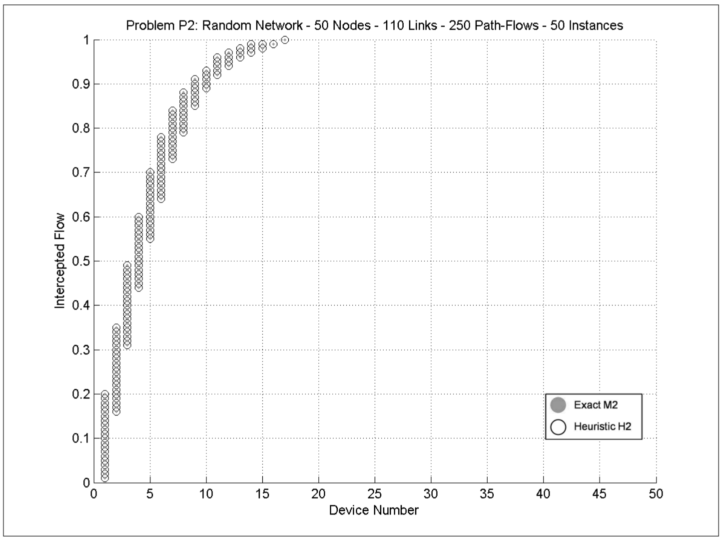


Figure 9.5: Problem P2: Model M2 solution vs. Heuristic H2 solutions over 50 instances.

ever, for a number of devices between *9* to *24*, a small distance can be perceived. Indeed in all cases we achieve more than *80%* of the total gain locating *14* devices (i.e. *29%* of the possible locations). Moreover, we obtain the maximum achievable gain with *50* devices, both for model and heuristics. This is due to two main reasons previously discussed. In fact we supposed that the a_{pi} coefficients were decreasing along a path and we assumed that each node was the origin of at least one path.

Figures (9.7) and (9.8) show the results achieved with the five different rules described in section (8.5) and (8.6), respectively for *ASH* and *TSH*. We can observe that in both cases the worst results are related to rule 1 and 2, whose maximum difference to the optimum is *0.0144* for *ASH* and *0.0143* for *TSH*. Instead the best results obtained with rule 5 are characterized by a maximum difference to the optimum equal to *0.0081* for *ASH* and equal to *0.002* for *TSH*. It is also important to note that in this case rule 3 returns better results than rules 1 and 2, but worse than rule 4, which is really efficient for problem *P3*, both in terms of quality of solution and computation time.

Figure (9.9) and (9.10) show the results obtained solving problem *P4* by *M4* and *H4*, respectively for one instance and *50* instances, plotting (on the X-axis) the number of devices as a function of the percentage of

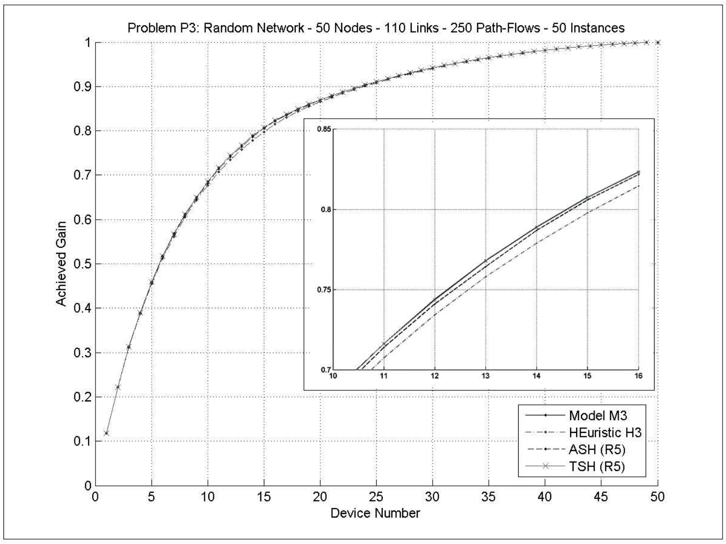


Figure 9.6: Problem P3: Model M3 solutions vs. Heuristic method solutions.

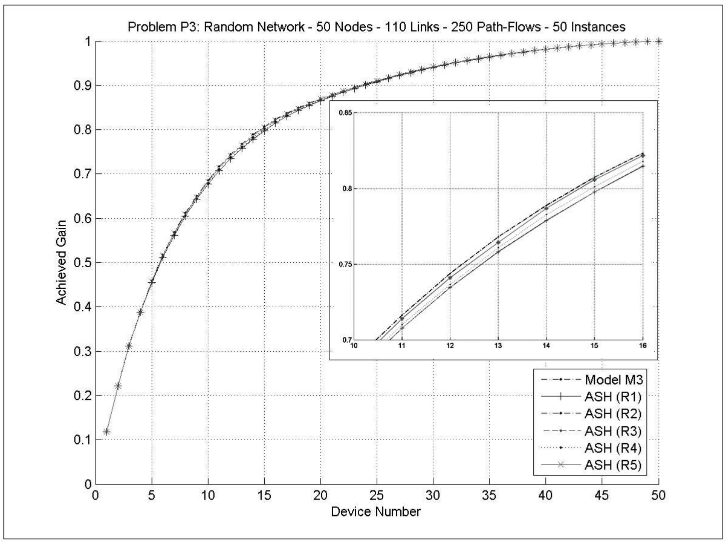


Figure 9.7: Problem P3: M3 solutions vs. ASH solutions (five rules).

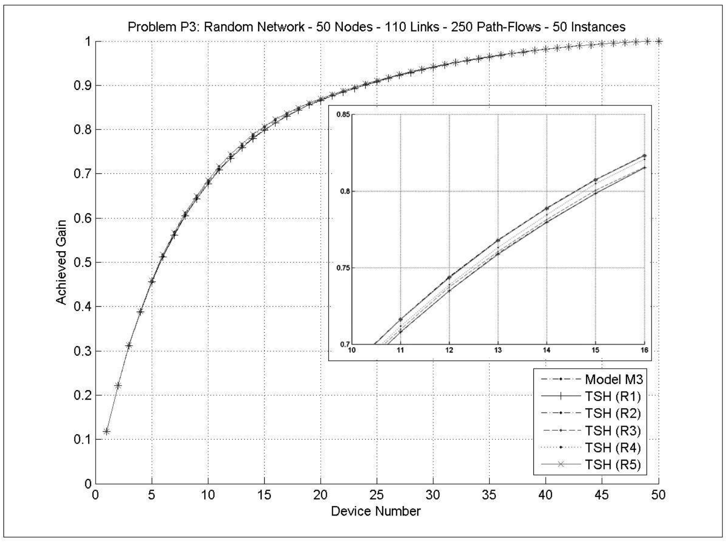


Figure 9.8: Problem P1: M3 solutions vs. TSH solutions (five rules).

the achieved gain ($(1-\epsilon)$, $0 \leq \epsilon < 1$ step size *0.01*). As regards figure (9.9) and (9.10) we can repeat the same observations already made for problem *P2*. In any case it is important to note that, contrary to what happened for problem *P2*, the results obtained for model *M4* do not exactly coincide with the ones obtained for *H4*, since this heuristic does not return the optimal solution. We can see, however, that in the worst case the heuristic provides a solution which uses at most two additional devices if compared to the optimal number.

9.3 Results of experimental tests

In this section we report the results of experimental tests run on random networks with size from *50* to *200* nodes for problem *P1* and *P3*. We will present some tables with results and computation times obtained solving problems by mathematical models and related heuristics. For problem *P1*, the values are obtained over *50* instances, setting m equal to *5%* of the number of nodes and varying the flow values in the range *1-30*. For problem *P3*, the values are obtained over *10* instances, setting m equal to *10%* of the number nodes and varying the flow values in the range *1-30*. For each problem, two tables will be provided, which

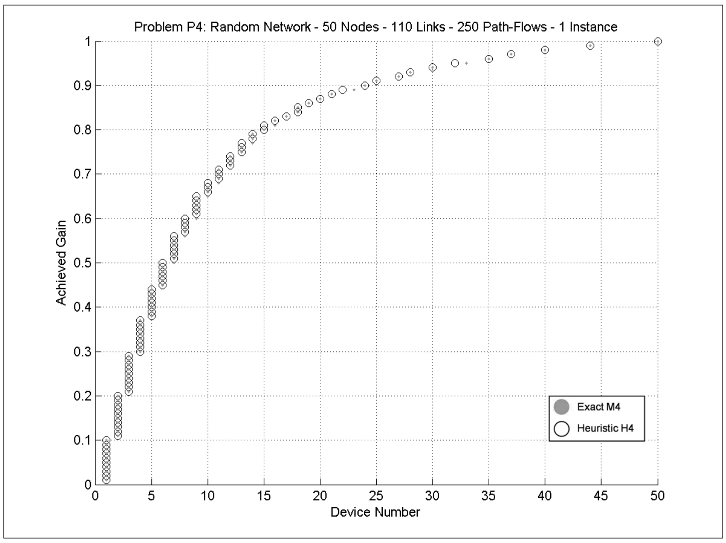


Figure 9.9: Problem P4: Model M4 solution vs. Heuristic H4 solutions over a single instance.

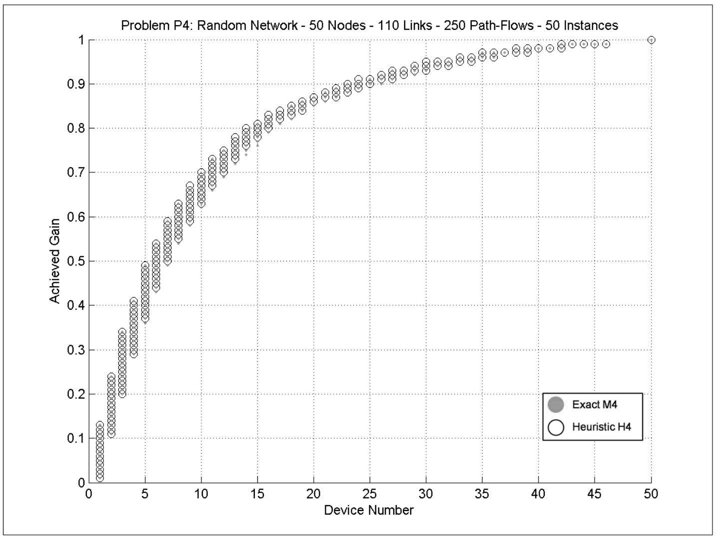


Figure 9.10: Problem P4: Model M4 solution vs. Heuristic H4 solutions over 50 instances.

respectively compare the results of the model and basic heuristics with the ones of the *ASH* and the *TSH* (using the five rules).
Table (9.2) reports the parameters used in tables 3–8.

# OptVal	Number of optimal solutions over all instances.
$z_i(Model)$	Objective function value with model for i .
$z_i(Heur)$	Objective function value with an heuristic method for i .
MeanVal	Average over all $i \in I$ of results with reference to optimal solutions:$[z_i(Heuristic)/z_i(Model)]/I, \forall i \in I$.
MaxVal	Maximum over all $i \in I$ of $[z_i(Heur)/z_i(Model)]$.
MinVal	Minimum over all $i \in I$ of $[z_i(Heur)/z_i(Model)]$.
MeanTime	Average over all $i \in I$ of the computation times *(s)*.
MaxTime	Maximum value over all $i \in I$ of computation times *(s)*.
MinTime	Minimum value over all $i \in I$ of computation times *(s)*.

Table 9.2: Test parameters.

Using the information in Tables 9.3 and 9.4 we can compute the difference between model and method solutions for the same instance I under investigation, as:

$$\Delta(z) = [1 - z(Heur_I)/z(Model_I)]$$

The largest difference is the one obtained with heuristic *H1* and are respectively: *0.0023*; *0.0013*; *0.0027* and *0.0035*. Concerning instead *ASH* and *TSH*, the best results in both cases are obtained with rule 5, and are respectively: *0.0009*; *0.0003*; *0.0001*; *0.0026* for *ASH* and *0*, *0*, *0* and *0.0002* for TSH. Therefore we can affirm that *TSH* provides solutions which are nearest to the optimum and in many cases it allows us to find exactly the optimal solution, but on the other hand its computation times increase very rapidly and sometimes they are higher than those of the model.

Using information in Tables 9.5 and 9.6 we see that the largest difference is the one obtained with the basic heuristic *H3* and its improvement and are respectively: *0.0082*; *0.0033*; *0.0013* and *0.0024*. Concerning instead *ASH* and *TSH*, the best results in both cases are obtained with rule 5, as with *P1*, and are respectively: *0.0046*; *0.0001*; *0.0013*; *0.0007* for *ASH* and *0*, *0*, *0* and *0.0003* for *TSH*. Even in this case the *TSH* heuristic provides solutions which are nearest to the optimum and in many cases it allows us to find exactly the optimal solution. Moreover, computation times of all heuristics are lower than those of the model.

	Model M1	HeurH1	H1Imp	ASH(R1)	ASH(R2)	ASH(R3)	ASH(R4)	ASH(R5)
Problem P1. Nodes: 50 (random). Paths: 250. Mean node number for paths: 4. Flows Range: 1-30.								
# OptVal	50	40	40	41	41	43	40	46
MinVal	-	0.970	0.970	0.970	0.970	0.970	0.970	0.970
MeanVal	-	0.998	0.998	0.998	0.998	0.998	0.998	0.999
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.109	0	0	0	0	0	0.970	0
MeanTime	0.213	0.002	0.003	0.003	0.003	0.004	0.998	0.008
MaxTime	0.937	0.016	0.016	0.016	0.016	0.016	1	0.016
Problem P1. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
# OptVal	50	35	35	35	35	37	35	44
MinVal	-	0.990	0.990	0.990	0.990	0.990	0.990	0.991
MeanVal	-	0.999	0.999	0.999	0.999	0.999	0.999	1.000
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.281	0	0	0.015	0.015	0.015	0.990	0.031
MeanTime	0.322	0.008	0.013	0.016	0.017	0.016	0.999	0.037
MaxTime	0.750	0.016	0.016	0.031	0.031	0.031	1	0.062
Problem P1. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
# OptVal	50	12	12	13	13	15	12	36
MinVal	-	0.989	0.989	0.989	0.989	0.989	0.989	0.989
MeanVal	-	0.997	0.997	0.997	0.998	0.997	0.997	0.999
MaxVal	-	1	1	1	1	1	1	1
MinTime	1.218	0.031	0.046	0.047	0.047	0.047	0.989	0.171
MeanTime	2.760	0.046	0.060	0.065	0.065	0.064	0.997	0.291
MaxTime	4.609	0.062	0.078	0.109	0.109	0.079	1	0.609
Problem P1. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
# OptVal	50	9	9	9	9	12	9	24
MinVal	-	0.989	0.989	0.989	0.989	0.989	0.989	0.989
MeanVal	-	0.997	0.997	0.997	0.997	0.997	0.997	0.997
MaxVal	-	1	1	1	1	1	1	1
MinTime	2.859	0.125	0.156	0.171	0.171	0.171	0.989	0.593
MeanTime	5.263	0.137	0.170	0.177	0.176	0.183	0.997	0.782
MaxTime	7.891	0.141	0.188	0.188	0.188	0.234	1	1.094
Problem P1. Nodes: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	9	8	14	17	16	9	37
MinVal	-	0.949	0.949	0.949	0.9553	0.949	0.949	0.9814
MeanVal	-	0.9835	0.9829	0.9869	0.9892	0.9884	0.9835	0.9973
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.109	0	0	0	0	0	0	0
MeanTime	0.1337	0	0.0025	0.0031	0.0025	0.0022	0.0047	0.0085
MaxTime	0.593	0	0.047	0.016	0.016	0.016	0.031	0.094
Problem P1. Nodes: 100 (mesh). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	5	5	5	5	5	5	7
MinVal	-	0.9663	0.9663	0.9663	0.9663	0.9663	0.9663	0.9663
MeanVal	-	0.9923	0.9923	0.9924	0.9924	0.9925	0.9923	0.9929
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.344	0	0	0.015	0.015	0.015	0.015	0.031
MeanTime	0.849	0.0143	0.0119	0.0206	0.0203	0.02	0.0225	0.041
MaxTime	1.875	0.016	0.016	0.032	0.032	0.032	0.032	0.094
Problem P1. Nodes: 144 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	5	5	5	5	5	5	16
MinVal	-	0.9837	0.9837	0.9837	0.9837	0.9837	0.9837	0.9865
MeanVal	-	0.994	0.994	0.994	0.994	0.9942	0.9942	0.9967
MaxVal	-	1	1	1	1	1	1	1
MinTime	3.312	0.031	0.046	0.078	0.078	0.078	0.078	0.14
MeanTime	6.2194	0.0447	0.054	0.0794	0.0816	0.0819	0.0854	0.3125
MaxTime	23.766	0.047	0.063	0.109	0.094	0.109	0.11	0.75
Problem P1. Nodes: 196 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	10	10	10	10	10	10	15
MinVal	-	0.9872	0.9872	0.9872	0.9872	0.9872	0.9875	0.9878
MeanVal	-	0.9956	0.9956	0.9956	0.9956	0.9956	0.9956	0.9972
MaxVal	-	1	1	1	1	1	1	1
MinTime	7.235	0.125	0.156	0.218	0.218	0.218	0.218	0.625
MeanTime	10.826	0.1373	0.1574	0.26	0.2493	0.2508	0.265	1.0938
MaxTime	20.031	0.141	0.172	0.329	0.329	0.391	0.406	2.64

Table 9.3: Problem P1: Model, basic heuristics and ASH on random and mesh networks. Facilities: 5%.

	Model M1	HeurH1	H1Imp	TSH(R1)	TSH(R2)	TSH(R3)	TSH(R4)	TSH(R5)
Problem P1. Node: 50 (random). Paths: 250. Mean node number for paths: 4. Flows Range: 1-30.								
# Opt Val	50	40	40	42	43	47	40	50
MinVal	-	0.970	0.970	0.970	0.970	0.993	0.970	1
MeanVal	-	0.998	0.998	0.998	0.999	1.000	0.998	1
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.109	0	0	0.031	0.046	0.031	0.062	0.062
MeanTime	0.213	0.002	0.003	0.047	0.047	0.039	0.066	0.076
MaxTime	0.937	0.016	0.016	0.062	0.047	0.047	0.079	0.079
Problem P1. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
# OptVal	50	35	35	36	36	38	35	50
MinVal	-	0.990	0.990	0.990	0.990	0.991	0.990	1
MeanVal	-	0.999	0.999	0.999	0.999	0.999	0.999	1
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.281	0	0	0.125	0.125	0.031	0.156	0.141
MeanTime	0.322	0.008	0.013	0.127	0.125	0.110	0.163	0.443
MaxTime	0.750	0.016	0.016	0.141	0.125	0.125	0.172	0.484
Problem P1. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
# OptVal	50	12	12	15	15	18	13	49
MinVal	-	0.989	0.989	0.989	0.989	0.991	0.989	0.999
MeanVal	-	0.997	0.997	0.998	0.998	0.998	0.997	1
MaxVal	-	1	1	1	1	1	1	1
MinTime	1.218	0.031	0.046	0.141	0.375	0.140	0.500	2.219
MeanTime	2.760	0.046	0.060	0.472	0.459	0.332	0.534	2.918
MaxTime	4.609	0.062	0.078	0.516	0.532	0.375	0.594	2.953
Problem P1. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
# OptVal	50	9	9	9	9	11	11	46
MinVal	-	0.989	0.989	0.989	0.989	0.989	0.989	0.993
MeanVal	-	0.997	0.997	0.997	0.997	0.997	0.997	1.000
MaxVal	-	1	1	1	1	1	1	1
MinTime	2.859	0.125	0.156	1.109	1.125	0.407	1.203	9.531
MeanTime	5.263	0.137	0.170	1.134	1.135	1.051	1.218	9.720
MaxTime	7.891	0.141	0.188	1.156	1.156	1.125	1.235	9.891
Problem P1. Node: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	9	8	26	39	33	9	50
MinVal	-	0.949	0.949	0.949	0.9801	0.9683	0.949	1
MeanVal	-	0.9835	0.9829	0.9926	0.9978	0.9959	0.9835	1
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.109	0	0	0.031	0.031	0.015	0.062	0.031
MeanTime	0.1337	0	0.0025	0.0462	0.0457	0.0397	0.0641	0.075
MaxTime	0.593	0	0.047	0.063	0.063	0.047	0.078	0.141
Problem P1. Nodes: 100 (mesh). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	5	5	5	5	16	5	44
MinVal	-	0.9663	0.9663	0.9663	0.9663	0.9663	0.9663	0.9899
MeanVal	-	0.9923	0.9923	0.9924	0.9924	0.9948	0.9923	0.9994
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.344	0	0	0.109	0.109	0.031	0.14	0.421
MeanTime	0.849	0.0143	0.0119	0.1175	0.1181	0.104	0.1503	0.4259
MaxTime	1.875	0.016	0.016	0.125	0.125	0.11	0.157	0.438
Problem P1. Nodes: 144 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	5	5	5	5	8	6	29
MinVal	-	0.9837	0.9837	0.9847	0.9847	0.9837	0.9837	0.9932
MeanVal	-	0.994	0.994	0.9941	0.9941	0.9946	0.9946	0.9988
MaxVal	-	1	1	1	1	1	1	1
MinTime	3.312	0.031	0.046	0.437	0.437	0.156	0.515	2.032
MeanTime	6.2194	0.0447	0.054	0.4462	0.4456	0.36	0.5337	2.5237
MaxTime	23.766	0.047	0.063	0.454	0.454	0.437	0.547	2.828
Problem P1. Nodes: 196 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	10	10	10	10	10	10	16
MinVal	-	0.9872	0.9872	0.9872	0.9872	0.9872	0.9875	0.9924
MeanVal	-	0.9956	0.9956	0.9956	0.9956	0.9959	0.996	0.9986
MaxVal	-	1	1	1	1	1	1	1
MinTime	7.235	0.125	0.156	1.14	1.125	0.422	1.25	2.906
MeanTime	10.826	0.1373	0.1574	1.3322	1.2816	1.0383	1.4609	10.2265
MaxTime	20.031	0.141	0.172	1.703	1.704	1.687	1.875	13.969

Table 9.4: Problem P1: Model, basic heuristics and TSH on random and mesh networks. Facilities: 5%.

	Model M3	HeurH3	H3Imp	ASH(R1)	ASH(R2)	ASH(R3)	ASH(R4)	ASH(R5)
Problem P3. Nodes: 50. Paths: 250 (random). Mean node number for paths: 4. Flows Range: 1-30.								
# OptVal	10	5	5	5	6	5	6	7
MinVal	-	0.971	0.971	0.971	0.971	0.971	0.973	0.973
MeanVal	-	0.992	0.992	0.992	0.993	0.992	0.995	0.995
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.188	0	0	0	0	0	0.973	0
MeanTime	0.350	0.008	0.003	0.013	0.013	0.012	0.995	0.014
MaxTime	1.218	0.016	0.015	0.016	0.016	0.016	1	0.016
Problem P3. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
# OptVal	10	5	5	5	5	5	6	9
MinVal	-	0.991	0.991	0.991	0.991	0.991	0.991	0.999
MeanVal	-	0.997	0.997	0.997	0.997	0.997	0.998	1.000
MaxVal	-	1	1	1	1	1	1	1
MinTime	1.656	0.047	0.046	0.046	0.047	0.047	0.991	0.109
MeanTime	1.911	0.050	0.049	0.052	0.055	0.053	0.998	0.130
MaxTime	2.110	0.062	0.063	0.063	0.063	0.063	1	0.172
Problem P3. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
# Opt Val	10	8	8	8	8	8	8	8
MinVal	-	0.990	0.990	0.990	0.990	0.990	0.990	0.990
MeanVal	-	0.999	0.999	0.999	0.999	0.999	0.999	0.999
MaxVal	-	1	1	1	1	1	1	1
MinTime	15.766	0.234	0.250	0.343	0.359	0.343	0.990	0.562
MeanTime	18.949	0.239	0.252	0.353	0.363	0.356	0.999	0.700
MaxTime	25.062	0.250	0.266	0.375	0.375	0.375	1	1.234
Problem P3. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
# OptVal	10	2	2	2	2	3	3	7
MinVal	-	0.993	0.993	0.993	0.993	0.993	0.993	0.996
MeanVal	-	0.998	0.998	0.998	0.998	0.998	0.998	0.999
MaxVal	-	1	1	1	1	1	1	1
MinTime	55.140	0.797	0.796	1.172	1.187	1.172	0.993	2.688
MeanTime	70.311	0.808	0.803	1.302	1.382	1.198	0.998	4.781
MaxTime	83.250	0.813	0.813	1.766	1.813	1.359	1	7.469
Problem P3. Nodes: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	3	3	3	3	3	4	4
MinVal	-	0.983	0.983	0.983	0.983	0.983	0.9867	0.9867
MeanVal	-	0.994	0.994	0.994	0.994	0.994	0.9957	0.9957
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.219	0	0	0	0	0	0	0
MeanTime	0.3251	0.0046	0.0064	0.0108	0.0141	0.0108	0.0125	0.0112
MaxTime	0.891	0.016	0.016	0.016	0.016	0.016	0.016	0.016
Problem P3. Nodes: 100 (mesh). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	3
MinVal	-	0.9874	0.9874	0.9874	0.9874	0.9874	0.9874	0.9902
MeanVal	-	0.9929	0.9929	0.9929	0.9929	0.9931	0.9929	0.9965
MaxVal	-	0.998	0.998	0.998	0.998	0.9981	0.998	1
MinTime	3.188	0.046	0.046	0.046	0.046	0.047	0.062	0.109
MeanTime	3.8549	0.0516	0.0546	0.0579	0.0608	0.0612	0.0624	0.1875
MaxTime	6.031	0.063	0.063	0.063	0.063	0.063	0.063	0.297
Problem P3. Nodes: 144 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	1
MinVal	-	0.9887	0.9887	0.9887	0.9887	0.9887	0.9887	0.9887
MeanVal	-	0.9929	0.9929	0.9929	0.9929	0.9929	0.9929	0.995
MaxVal	-	0.996	0.996	0.996	0.996	0.996	0.996	1
MinTime	22.438	0.218	0.218	0.343	0.344	0.343	0.344	0.562
MeanTime	29.1236	0.2343	0.2297	0.347	0.3609	0.3485	0.3565	1.0579
MaxTime	39.719	0.25	0.235	0.36	0.375	0.375	0.375	2.047
Problem P3. Nodes: 196 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	1
MinVal	-	0.9963	0.9963	0.9972	0.9972	0.9963	0.9963	0.9977
MeanVal	-	0.9982	0.9982	0.9984	0.9984	0.9982	0.9982	0.999
MaxVal	-	0.9995	0.9995	0.9999	0.9999	0.9995	0.9995	1
MinTime	104.031	0.828	0.781	1.219	1.265	1.203	1.218	2.765
MeanTime	136.398	0.9875	0.7938	1.5451	1.589	1.5063	1.4749	5.8405
MaxTime	182.344	1.125	0.797	2.422	2.157	1.813	1.937	13.313

Table 9.5: Problem P3: Model, basic heuristics and ASH on random and mesh networks. Facilities: 10%.

	Model M3	HeurH3	H3Imp	TSH(R1)	TSH(R2)	TSH(R3)	TSH(R4)	TSH(R5)
Problem P3. Nodes: 50 (random). Paths: 250. Mean node number for paths: 4. Flows Range: 1-30.								
# OptVal	10	5	5	7	6	8	8	10
MinVal	-	0.971	0.971	0.971	0.971	0.971	0.992	1
MeanVal	-	0.992	0.992	0.993	0.993	0.994	0.999	1
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.188	0	0	0.031	0.047	0.031	0.063	0.109
MeanTime	0.350	0.008	0.003	0.045	0.055	0.038	0.077	0.109
MaxTime	1.218	0.016	0.015	0.047	0.063	0.047	0.079	0.109
Problem P3. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
# OptVal	10	5	5	5	5	6	9	10
MinVal	-	0.991	0.991	0.991	0.991	0.991	0.999	1
MeanVal	-	0.997	0.997	0.997	0.997	0.998	1.000	1
MaxVal	-	1	1	1	1	1	1	1
MinTime	1.656	0.047	0.046	0.187	0.235	0.172	0.265	1.203
MeanTime	1.911	0.050	0.049	0.198	0.247	0.183	0.269	1.219
MaxTime	2.110	0.062	0.063	0.204	0.250	0.188	0.282	1.235
Problem P3. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
# OptVal	10	8	8	8	8	8	9	10
MinVal	-	0.990	0.990	0.990	0.990	0.990	0.997	1
MeanVal	-	0.999	0.999	0.999	0.999	0.999	1.000	1
MaxVal	-	1	1	1	1	1	1	1
MinTime	15.766	0.234	0.250	0.984	1.109	0.828	1.156	6.937
MeanTime	18.949	0.239	0.252	0.991	1.122	0.947	1.175	8.198
MaxTime	25.062	0.250	0.266	1	1.125	0.969	1.188	9.469
Problem P3. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
# OptVal	10	0	0	0	0	0	0	2
MinVal	-	0.991	0.991	0.991	0.991	0.991	0.992	0.996
MeanVal	-	0.993	0.993	0.993	0.993	0.993	0.995	0.999
MaxVal	-	0.995	0.995	0.995	0.995	0.995	0.998	1
MinTime	55.140	0.797	0.796	2.781	1.687	2.718	3.094	25.781
MeanTime	70.311	0.808	0.803	3.178	3.380	2.736	3.475	33.181
MaxTime	83.250	0.813	0.813	4.141	4.563	2.766	4.657	38.219
Problem P3. Nodes: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	3	3	3	3	6	5	10
MinVal	-	0.983	0.983	0.983	0.983	0.983	0.9867	1
MeanVal	-	0.994	0.994	0.994	0.994	0.9959	0.9969	1
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.219	0	0	0.032	0.047	0.031	0.062	0.093
MeanTime	0.3251	0.0046	0.0064	0.0455	0.0547	0.0392	0.075	0.1106
MaxTime	0.891	0.016	0.016	0.047	0.063	0.047	0.079	0.125
Problem P3. Nodes: 100 (mesh). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	8
MinVal	-	0.9874	0.9874	0.9874	0.9874	0.9902	0.9874	0.9947
MeanVal	-	0.9929	0.9929	0.9929	0.9929	0.9944	0.9929	0.9994
MaxVal	-	0.998	0.998	0.998	0.998	0.998	0.998	1
MinTime	3.188	0.046	0.046	0.203	0.25	0.093	0.265	1.25
MeanTime	3.8549	0.0516	0.0546	0.2062	0.2641	0.1732	0.275	1.2686
MaxTime	6.031	0.063	0.063	0.219	0.266	0.203	0.282	1.281
Problem P3. Nodes: 144 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	5
MinVal	-	0.9887	0.9887	0.9887	0.9887	0.9887	0.9887	0.9951
MeanVal	-	0.9929	0.9929	0.9929	0.9929	0.9931	0.9935	0.9988
MaxVal	-	0.996	0.996	0.996	0.996	0.996	0.9984	1
MinTime	22.438	0.218	0.218	0.5	0.562	0.5	1.14	6.75
MeanTime	29.1236	0.2343	0.2297	0.8779	1.1046	0.7813	1.156	8.0078
MaxTime	39.719	0.25	0.235	0.985	1.187	0.938	1.172	9.219
Problem P3. Nodes: 196 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	4
MinVal	-	0.9963	0.9963	0.9972	0.9972	0.9972	0.9963	0.999
MeanVal	-	0.9982	0.9982	0.9984	0.9984	0.9983	0.9983	0.9998
MaxVal	-	0.9995	0.9995	0.9999	0.9999	0.9995	0.9995	1
MinTime	104.031	0.828	0.781	2.844	3.313	1.625	3.187	33.625
MeanTime	136.398	0.9875	0.7938	3.9063	4.4861	2.9389	4.2359	42.7938
MaxTime	182.344	1.125	0.797	6.016	6.657	5.921	6.266	54.75

Table 9.6: Problem P3: Model, basic heuristics and TSH on random and mesh networks. Facilities: 10%.

	Model M1	HeurH1	H1Imp	ASH(R1)	ASH(R2)	ASH(R3)	ASH(R4)	ASH(R5)
Problem P1. Nodes: 50 (random). Paths: 250. Mean node number for paths: 4. Flows Range: 1-30.								
#OptVal	-	27	27	28	28	29	27	41
MinVal	-	0.940	0.940	0.940	0.940	0.941	0.940	0.961
MeanVal	-	0.991	0.991	0.991	0.991	0.993	0.991	0.998
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	0.109	0.000	0.000	0.000	0.000	0.000	0.000	0.000
MeanTime	0.218	0.002	0.004	0.004	0.005	0.005	0.007	0.013
MaxTime	0.469	0.016	0.016	0.016	0.016	0.016	0.016	0.031
Problem P1. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	31	31	31	31	31	36	37
MinVal	-	0.989	0.989	0.989	0.989	0.989	0.989	0.992
MeanVal	-	0.998	0.998	0.998	0.998	0.998	0.999	0.999
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	0.265	0.015	0.015	0.015	0.015	0.015	0.015	0.062
MeanTime	0.783	0.020	0.021	0.025	0.024	0.025	0.032	0.091
MaxTime	2.203	0.032	0.032	0.032	0.032	0.032	0.047	0.235
Problem P1. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	10	10	10	10	10	11	28
MinVal	-	0.978	0.978	0.979	0.978	0.978	0.982	0.988
MeanVal	-	0.996	0.996	0.996	0.996	0.996	0.997	0.998
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	3.187	0.078	0.094	0.093	0.093	0.093	0.109	0.406
MeanTime	9.497	0.085	0.106	0.109	0.111	0.109	0.127	0.965
MaxTime	35.891	0.094	0.125	0.141	0.141	0.140	0.250	2.313
Problem P1. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	2	11
MinVal	-	0.992	0.992	0.992	0.992	0.992	0.992	0.993
MeanVal	-	0.996	0.996	0.996	0.996	0.996	0.996	0.998
MaxVal	-	0.999	0.999	0.999	0.999	0.999	1.000	1.000
MinTime	9.422	0.250	0.312	0.328	0.328	0.328	0.328	1.781
MeanTime	20.381	0.266	0.327	0.340	0.339	0.340	0.374	4.746
MaxTime	71.219	0.282	0.344	0.344	0.344	0.406	0.547	12.609
Problem P1. Nodes: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	3	3	3	3	5	4	17
MinVal	-	0.950	0.950	0.950	0.950	0.950	0.950	0.960
MeanVal	-	0.981	0.980	0.981	0.981	0.982	0.981	0.992
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	0.109	0.000	0.000	0.000	0.000	0.000	0.000	0.000
MeanTime	0.159	0.005	0.003	0.004	0.004	0.004	0.006	0.017
MaxTime	0.312	0.031	0.016	0.016	0.016	0.016	0.016	0.032
Problem P1. Nodes: 100 (mesh). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	2
MinVal	-	0.974	0.974	0.974	0.974	0.974	0.974	0.987
MeanVal	-	0.991	0.991	0.991	0.991	0.991	0.991	0.994
MaxVal	-	0.999	0.999	0.999	0.999	0.999	0.999	1.000
MinTime	0.297	0.000	0.015	0.031	0.031	0.016	0.031	0.063
MeanTime	0.715	0.024	0.019	0.040	0.039	0.038	0.044	0.164
MaxTime	2.000	0.032	0.032	0.047	0.047	0.047	0.047	0.594
Problem P1. Nodes: 144 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	1	1	1	1	1	1	5
MinVal	-	0.982	0.982	0.982	0.982	0.982	0.982	0.985
MeanVal	-	0.994	0.994	0.994	0.994	0.994	0.994	0.996
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	1.656	0.093	0.125	0.125	0.125	0.125	0.140	0.468
MeanTime	12.088	0.099	0.134	0.139	0.149	0.150	0.160	1.155
MaxTime	38.844	0.110	0.141	0.203	0.235	0.219	0.250	2.657
Problem P1. Nodes: 196 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.988	0.988	0.988	0.988	0.988	0.988	0.993
MeanVal	-	0.993	0.993	0.994	0.994	0.993	0.994	0.997
MaxVal	-	0.998	0.998	0.998	0.998	0.998	0.998	1.000
MinTime	7.781	0.312	0.390	0.390	0.390	0.390	0.406	1.640
MeanTime	22.013	0.328	0.408	0.457	0.459	0.451	0.493	6.964
MaxTime	83.797	0.344	0.422	0.875	0.687	0.687	0.735	15.984

Table 9.7: Problem P1: Model, basic heuristics and ASH on random and mesh networks. Facilities: 10%.

	Model M1	HeurH1	H1Imp	TSH(R1)	TSH(R2)	TSH(R3)	TSH(R4)	TSH(R5)
Problem P1. Node: 50 (random). Paths: 250. Mean node number for paths: 4. Flows Range: 1-30.								
#OptVal	-	27	27	31	33	37	33	50
MinVal	-	0.940	0.940	0.940	0.940	0.952	0.941	1.000
MeanVal	-	0.991	0.991	0.992	0.993	0.996	0.995	1.000
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	0.109	0.000	0.000	0.015	0.046	0.015	0.078	0.031
MeanTime	0.218	0.002	0.004	0.048	0.049	0.039	0.082	0.110
MaxTime	0.469	0.016	0.016	0.063	0.063	0.047	0.094	0.125
Problem P1. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	31	31	31	31	33	36	48
MinVal	-	0.989	0.989	0.989	0.989	0.989	0.992	0.997
MeanVal	-	0.998	0.998	0.998	0.998	0.999	0.999	1.000
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	0.265	0.015	0.015	0.156	0.156	0.047	0.234	0.703
MeanTime	0.783	0.020	0.021	0.167	0.165	0.140	0.240	1.161
MaxTime	2.203	0.032	0.032	0.172	0.172	0.157	0.250	1.219
Problem P1. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	10	10	10	10	10	14	41
MinVal	-	0.978	0.978	0.984	0.984	0.984	0.989	0.996
MeanVal	-	0.996	0.996	0.996	0.996	0.996	0.998	1.000
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	3.187	0.078	0.094	0.219	0.219	0.203	0.687	2.047
MeanTime	9.497	0.085	0.106	0.569	0.564	0.494	0.697	9.046
MaxTime	35.891	0.094	0.125	0.594	0.579	0.578	0.704	9.453
Problem P1. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	4	23
MinVal	-	0.992	0.992	0.992	0.992	0.992	0.993	0.995
MeanVal	-	0.996	0.996	0.996	0.996	0.996	0.997	0.999
MaxVal	-	0.999	0.999	0.999	0.999	1.000	1.000	1.000
MinTime	9.422	0.250	0.312	0.719	0.719	0.703	2.062	29.625
MeanTime	20.381	0.266	0.327	1.734	1.742	1.662	2.084	31.306
MaxTime	71.219	0.282	0.344	1.875	1.844	1.828	2.140	31.750
	Model M1	HeurH1	H1Imp	TSH(R1)	TSH(R2)	TSH(R3)	TSH(R4)	TSH(R5)
Problem P1. Node: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	3	3	3	3	12	5	35
MinVal	-	0.950	0.950	0.950	0.950	0.960	0.950	0.972
MeanVal	-	0.981	0.980	0.981	0.981	0.989	0.982	0.998
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	0.109	0.000	0.000	0.031	0.031	0.031	0.063	0.093
MeanTime	0.159	0.005	0.003	0.046	0.047	0.038	0.077	0.103
MaxTime	0.312	0.031	0.016	0.047	0.047	0.047	0.079	0.110
Problem P1. Nodes: 100 (mesh). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	13
MinVal	-	0.974	0.974	0.974	0.974	0.977	0.974	0.989
MeanVal	-	0.991	0.991	0.991	0.991	0.992	0.992	0.996
MaxVal	-	0.999	0.999	0.999	0.999	1.000	1.000	1.000
MinTime	0.297	0.000	0.015	0.078	0.203	0.062	0.296	1.062
MeanTime	0.715	0.024	0.019	0.206	0.208	0.176	0.302	1.383
MaxTime	2	0.032	0.032	0.219	0.219	0.203	0.313	1.500
Problem P1. Nodes: 144 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	1	1	1	1	1	2	9
MinVal	-	0.982	0.982	0.989	0.989	0.984	0.982	0.994
MeanVal	-	0.994	0.994	0.994	0.994	0.994	0.995	0.998
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	1.656	0.093	0.125	0.250	0.250	0.250	0.812	2.312
MeanTime	12.088	0.099	0.134	0.651	0.689	0.583	0.904	7.830
MaxTime	38.844	0.110	0.141	0.938	0.953	0.907	1.204	11.171
Problem P1. Nodes: 196 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	4
MinVal	-	0.988	0.988	0.988	0.988	0.988	0.988	0.993
MeanVal	-	0.993	0.993	0.994	0.994	0.994	0.994	0.998
MaxVal	-	0.998	0.998	0.998	0.998	0.998	0.998	1.000
MinTime	7.781	0.312	0.390	0.734	0.718	0.719	2.000	12.125
MeanTime	22.013	0.328	0.408	1.779	1.874	1.508	2.291	31.598
MaxTime	83797	0.344	0.422	2.547	2.532	2.484	2.985	35.640

Table 9.8: Problem P1: Model, basic heuristics and TSH on random and mesh networks. Facilities: 10%.

	Model M3	HeurH3	H3Imp	ASH(R1)	ASH(R2)	ASH(R3)	ASH(R4)	ASH(R5)
Problem P3. Nodes: 50 (random). Paths: 250. Mean node number for paths: 4. Flows Range: 1-30.								
#OptVal	-	1	1	1	1	1	1	7
MinVal	-	0,9757	0,9757	0,9757	0,9757	0,9757	0,9801	0,9924
MeanVal	-	0,9895	0,9895	0.9895	0.9895	0.9895	0.9931	0.9991
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.156	0.015	0	0.016	0.016	0.015	0.015	0.031
MeanTime	0.2343	0.0156	0.0141	0.0205	0.022	0.0217	0.0232	0.0562
MaxTime	0.594	0.016	0.016	0.031	0.031	0.031	0.046	0.094
Problem P3. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	1
MinVal	-	0.9906	0.9906	0.9906	0.9906	0.9906	0.9906	0.9919
MeanVal	-	0.9936	0.9936	0.9939	0.9937	0.9936	0.9942	0.9968
MaxVal	-	0.9987	0.9987	0.9987	0.9987	0.9987	0.9987	1
MinTime	1.218	0.156	0.109	0.234	0.25	0.234	0.25	0.375
MeanTime	1.8125	0.1702	0.1171	0.2469	0.25	0.2388	0.2579	0.897
MaxTime	6.141	0.172	0.125	0.265	0.25	0.25	0.282	1.86
Problem P3. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9962	0.9962	0.9962	0.9962	0.9962	0.9962	0.9981
MeanVal	-	0.9983	0.9983	0.9983	0.9983	0.9983	0.9983	0.999
MaxVal	-	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998
MinTime	7.344	1.328	0.812	1.375	1.39	1.359	1.406	3.25
MeanTime	10.5375	1.3311	0.8141	1.5827	1.5296	1.5561	1.5655	6.3784
MaxTime	15.875	1.344	0.828	2.079	2.094	2.062	2.109	16.187
Problem P3. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	1
MinVal	-	0.9909	0.9909	0.9909	0.9909	0.9909	0.9909	0.9964
MeanVal	-	0.9929	0.9929	0.9929	0.9929	0.9929	0.994	0.9982
MaxVal	-	0.9952	0.9952	0.9952	0.9952	0.9952	0.9971	1
MinTime	31.406	4.781	2.562	4.953	4.969	4.953	5.016	27.438
MeanTime	56.1796	4.8046	2.5627	5.4405	5.711	5.7187	5.9238	79.1829
MaxTime	121.64	4.844	2.563	7.344	7.453	7.344	7.782	115.234
Problem P3. Nodes: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	1	1	1	1	1	2	7
MinVal	-	0.9816	0.9816	0.9816	0.9816	0.9816	0.9833	0.9924
MeanVal	-	0.9894	0.9894	0.9894	0.9898	0.9901	0.9916	0.9986
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.187	0.015	0	0.016	0.015	0.015	0.015	0.031
MeanTime	0.3282	0.0202	0.0095	0.0207	0.0217	0.0204	0.0281	0.0624
MaxTime	0.968	0.032	0.016	0.032	0.031	0.031	0.032	0.078
Problem P3. Nodes: 100 (mesh). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	2
MinVal	-	0.9919	0.9919	0.9919	0.9919	0.9919	0.9928	0.9954
MeanVal	-	0.996	0.996	0.996	0.996	0.996	0.9965	0.998
MaxVal	-	0.9991	0.9991	0.9991	0.9991	0.9991	0.9996	1
MinTime	3.515	0.171	0.109	0.25	0.25	0.25	0.25	0.391
MeanTime	9.2905	0.1765	0.1157	0.25	0.2563	0.2547	0.2766	1.0486
MaxTime	18.828	0.188	0.125	0.25	0.266	0.266	0.328	2.141
Problem P3. Nodes: 144 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9937	0.9937	0.9937	0.9937	0.994	0.9942	0.9964
MeanVal	-	0.9954	0.9954	0.9954	0.9954	0.9955	0.9958	0.9976
MaxVal	-	0.998	0.998	0.998	0.998	0.998	0.998	0.9992
MinTime	26.5	1.234	0.718	1.282	1.297	1.281	1.328	6.5
MeanTime	47.3362	1.2437	0.7313	1.6642	1.5579	1.4862	1.6172	11.9187
MaxTime	93.75	1.25	0.735	1.938	1.953	1.938	2.109	17.344
Problem P3. Nodes: 196 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9945	0.9945	0.9945	0.9945	0.9945	0.995	0.9955
MeanVal	-	0.9957	0.9957	0.9958	0.9957	0.9957	0.9961	0.9971
MaxVal	-	0.997	0.997	0.997	0.997	0.997	0.9974	0.9987
MinTime	52.297	5.312	2.468	5.891	5.844	5.843	7.047	24.719
MeanTime	79.0748	6.4297	2.7936	7.0875	7.2156	6.9813	8.65	75.497
MaxTime	115.407	9.093	3.609	9.687	10.297	10.235	11.86	194.813

Table 9.9: Problem P3: Model, basic heuristics and ASH on random and mesh networks. Facilities: 25%.

	Model M3	HeurH3	H3Imp	TSH(R1)	TSH(R2)	TSH(R3)	TSH(R4)	TSH(R5)
Problem P3. Nodes: 50 (random). Paths: 250. Mean node number for paths: 4. Flows Range: 1-30.								
#OptVal	-	1	1	1	1	1	2	7
MinVal	-	0,9816	0,9816	0,9816	0,9816	0,9816	0,9833	0,9924
MeanVal	-	0.9894	0.9894	0.9894	0.9898	0.9901	0.9916	0.9986
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.187	0.015	0	0.016	0.015	0.015	0.015	0.031
MeanTime	0.3282	0.0202	0.0095	0.0207	0.0217	0.0204	0.0281	0.0624
MaxTime	0.968	0.032	0.016	0.032	0.031	0.031	0.032	0.078
Problem P3. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	2
MinVal	-	0.9919	0.9919	0.9919	0.9919	0.9919	0.9928	0.9954
MeanVal	-	0.996	0.996	0.996	0.996	0.996	0.9965	0.998
MaxVal	-	0.9991	0.9991	0.9991	0.9991	0.9991	0.9996	1
MinTime	3.515	0.171	0.109	0.25	0.25	0.25	0.25	0.391
MeanTime	9.2905	0.1765	0.1157	0.25	0.2563	0.2547	0.2766	1.0486
MaxTime	18.828	0.188	0.125	0.25	0.266	0.266	0.328	2.141
Problem P3. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9937	0.9937	0.9937	0.9937	0.994	0.9942	0.9964
MeanVal	-	0.9954	0.9954	0.9954	0.9954	0.9955	0.9958	0.9976
MaxVal	-	0.998	0.998	0.998	0.998	0.998	0.998	0.9992
MinTime	26.5	1.234	0.718	1.282	1.297	1.281	1.328	6.5
MeanTime	47.3362	1.2437	0.7313	1.6642	1.5579	1.4862	1.6172	11.9187
MaxTime	93.75	1.25	0.735	1.938	1.953	1.938	2.109	17.344
Problem P3. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9945	0.9945	0.9945	0.9945	0.9945	0.995	0.9955
MeanVal	-	0.9957	0.9957	0.9958	0.9957	0.9957	0.9961	0.9971
MaxVal	-	0.997	0.997	0.997	0.997	0.997	0.9974	0.9987
MinTime	52.297	5.312	2.468	5.891	5.844	5.843	7.047	24.719
MeanTime	79.0748	6.4297	2.7936	7.0875	7.2156	6.9813	8.65	75.497
MaxTime	115.407	9.093	3.609	9.687	10.297	10.235	11.86	194.813
Problem P3. Nodes: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	1	1	1	1	2	3	8
MinVal	-	0.9816	0.9816	0.9816	0.9816	0.9849	0.9852	0.9983
MeanVal	-	0.9894	0.9894	0.9896	0.9898	0.9928	0.9953	0.9998
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.187	0.015	0	0.031	0.031	0.031	0.14	0.281
MeanTime	0.3282	0.0202	0.0095	0.0558	0.0735	0.0484	0.1422	0.2876
MaxTime	0.968	0.032	0.016	0.062	0.079	0.063	0.156	0.297
Problem P3. Nodes: 100 (mesh). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	4
MinVal	-	0.9919	0.9919	0.9919	0.9919	0.9919	0.9947	0.9981
MeanVal	-	0.996	0.996	0.996	0.996	0.996	0.9975	0.9996
MaxVal	-	0.9991	0.9991	0.9991	0.9991	0.9991	1	1
MinTime	3.515	0.171	0.109	0.312	0.656	0.5	0.796	3.031
MeanTime	9.2905	0.1765	0.1157	0.503	0.6688	0.5032	0.8013	4.8561
MaxTime	18.828	0.188	0.125	0.532	0.687	0.516	0.812	6.234
Problem P3. Nodes: 144 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9937	0.9937	0.994	0.994	0.9939	0.9942	0.9974
MeanVal	-	0.9954	0.9954	0.9956	0.9955	0.9955	0.9965	0.9987
MaxVal	-	0.998	0.998	0.9985	0.998	0.998	0.9987	0.9998
MinTime	26.5	1.234	0.718	2.218	2.593	1.531	2.89	21.922
MeanTime	47.3362	1.2437	0.7313	2.7142	3.1045	2.2921	3.3564	39.6781
MaxTime	93.75	1.25	0.735	3.328	3.875	3.25	4.313	44.063
Problem P3. Nodes: 196 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9945	0.9945	0.9945	0.9945	0.9945	0.995	0.9961
MeanVal	-	0.9957	0.9957	0.9958	0.9957	0.9957	0.9962	0.9976
MaxVal	-	0.997	0.997	0.997	0.997	0.997	0.9974	0.9989
MinTime	52.297	5.312	2.468	8.062	7.344	8.047	13.829	87.141
MeanTime	79.0748	6.4297	2.7936	10.4719	12.4749	10.6937	14.8985	191.698
MaxTime	115.407	9.093	3.609	12.407	17.812	16.719	17.235	281.469

Table 9.10: Problem P3: Model, basic heuristics and TSH on random and mesh networks. Facilities: 25%.

	Model M1	HeurH1	H1Imp	ASH(R1)	ASH(R2)	ASH(R3)	ASH(R4)	ASH(R5)
Problem P1. Nodes: 50 (random). Paths: 250. Mean node number for paths: 4. Flows Range: 1-30.								
#OptVal	-	3	3	3	3	3	3	7
MinVal	-	0.980	0.982	0.980	0.980	0.980	0.980	0.980
MeanVal	-	0.992	0.993	0.992	0.992	0.992	0.992	0.994
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	0.093	0.000	0.000	0.000	0.000	0.000	0.000	0.015
MeanTime	0.359	0.005	0.008	0.008	0.008	0.007	0.013	0.031
MaxTime	0.593	0.016	0.016	0.016	0.016	0.016	0.016	0.094
Problem P1. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	2	2	2	2	2	2	6
MinVal	-	0.988	0.988	0.988	0.988	0.988	0.988	0.988
MeanVal	-	0.996	0.996	0.996	0.996	0.996	0.997	0.998
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	0.391	0.031	0.046	0.032	0.046	0.031	0.047	0.250
MeanTime	1.734	0.039	0.053	0.050	0.051	0.049	0.067	0.815
MaxTime	2.641	0.047	0.063	0.063	0.078	0.078	0.110	2.516
Problem P1. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	5
MinVal	-	0.993	0.993	0.993	0.993	0.993	0.994	0.996
MeanVal	-	0.997	0.997	0.997	0.997	0.997	0.998	0.999
MaxVal	-	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MinTime	3.407	0.188	0.250	0.250	0.250	0.250	0.266	2.046
MeanTime	17.294	0.208	0.256	0.264	0.265	0.262	0.350	6.859
MaxTime	39.063	0.219	0.266	0.359	0.360	0.359	0.593	13.281
Problem P1. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.990	0.990	0.990	0.990	0.990	0.991	0.994
MeanVal	-	0.993	0.993	0.993	0.993	0.993	0.994	0.997
MaxVal	-	0.997	0.997	0.997	0.997	0.997	0.997	1.000
MinTime	27.687	0.625	0.765	0.750	0.750	0.766	0.797	22.343
MeanTime	182.670	0.643	0.782	0.798	0.791	0.800	1.017	66.451
MaxTime	765.640	0.672	0.797	0.937	0.938	0.922	1.516	185.120
Problem P1. Nodes: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	6
MinVal	-	0.974	0.980	0.974	0.974	0.974	0.980	0.983
MeanVal	-	0.989	0.989	0.989	0.989	0.989	0.990	0.994
MaxVal	-	0.997	0.997	0.997	0.997	0.997	0.998	1.000
MinTime	0.094	0.000	0.000	0.000	0.000	0.000	0.000	0.015
MeanTime	0.153	0.004	0.010	0.007	0.007	0.005	0.012	0.037
MaxTime	0.359	0.016	0.016	0.016	0.016	0.016	0.031	0.079
Problem P1. Nodes: 100 (mesh). Paths:1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	3
MinVal	-	0.990	0.990	0.990	0.990	0.990	0.993	0.994
MeanVal	-	0.995	0.995	0.995	0.995	0.995	0.996	0.998
MaxVal	-	0.997	0.998	0.997	0.997	0.997	0.998	1.000
MinTime	0.203	0.047	0.062	0.062	0.062	0.062	0.078	0.593
MeanTime	1.201	0.064	0.072	0.076	0.077	0.075	0.115	1.379
MaxTime	2.640	0.156	0.079	0.110	0.110	0.094	0.235	3.625
Problem P1. Nodes: 144 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	1
MinVal	-	0.992	0.992	0.992	0.992	0.992	0.992	0.995
MeanVal	-	0.995	0.995	0.995	0.995	0.995	0.996	0.998
MaxVal	-	0.998	0.999	0.998	0.998	0.998	0.999	1.000
MinTime	0.829	0.250	0.297	0.296	0.281	0.281	0.312	3.313
MeanTime	3.169	0.280	0.335	0.319	0.320	0.336	0.480	12.590
MaxTime	9.750	0.375	0.453	0.437	0.438	0.500	0.797	30.125
Problem P1. Nodes: 196 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.992	0.992	0.992	0.992	0.992	0.993	0.995
MeanVal	-	0.995	0.995	0.995	0.995	0.995	0.995	0.998
MaxVal	-	0.996	0.997	0.996	0.996	0.996	0.998	1.000
MinTime	4.500	0.781	0.968	0.921	0.906	0.906	0.984	40.297
MeanTime	19.084	0.797	0.986	1.099	1.057	1.041	1.609	93.686
MaxTime	87.735	0.813	1.016	1.531	1.375	1.703	2.968	185.320

Table 9.11: Problem P1: Model, basic heuristics and ASH on random and mesh networks. Facilities: 25%.

	Model M1	HeurH1	H1Imp	TSH(R1)	TSH(R2)	TSH(R3)	TSH(R4)	TSH(R5)
Problem P1. Node: 50 (random). Paths: 250. Mean node number for paths: 4. Flows Range: 1-30.								
#OptVal	-	3	3	3	3	4	6	31
MinVal	-	0.9801	0.982	0.9801	0.9801	0.9801	0.9801	0.9899
MeanVal	-	0.9922	0.9925	0.9925	0.9923	0.9931	0.9937	0.9982
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.093	0	0	0	0.015	0.016	0.14	0.125
MeanTime	0.3593	0.0053	0.0075	0.0456	0.0459	0.0422	0.1406	0.2919
MaxTime	0.593	0.016	0.016	0.063	0.063	0.047	0.141	0.297
Problem P1. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	2	2	2	2	2	4	18
MinVal	-	0.9881	0.9881	0.9895	0.9909	0.9881	0.9909	0.9972
MeanVal	-	0.9963	0.9963	0.9965	0.9965	0.9965	0.9976	0.9994
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.391	0.031	0.046	0.078	0.093	0.094	0.453	1.969
MeanTime	1.7344	0.0388	0.0525	0.2084	0.1754	0.2209	0.4597	5.4515
MaxTime	2.641	0.047	0.063	0.265	0.25	0.235	0.469	5.704
Problem P1. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	18
MinVal	-	0.9932	0.9932	0.9935	0.9935	0.9932	0.9956	0.9976
MeanVal	-	0.9968	0.9969	0.997	0.9971	0.9968	0.9981	0.9996
MaxVal	-	0.9995	0.9995	0.9995	0.9995	0.9997	0.9998	1
MinTime	3.407	0.188	0.25	0.5	0.5	0.5	1.765	15.172
MeanTime	17.294	0.2081	0.2563	1.2059	1.1518	1.1146	1.7813	36.474
MaxTime	39.063	0.219	0.266	1.25	1.25	1.204	1.797	37.969
Problem P1. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.99	0.99	0.99	0.99	0.9905	0.9905	0.994
MeanVal	-	0.9934	0.9934	0.9935	0.9934	0.9935	0.9949	0.9982
MaxVal	-	0.9966	0.9966	0.9966	0.9966	0.9966	0.999	0.9998
MinTime	27.687	0.625	0.765	1.484	1.484	1.453	4.156	80.984
MeanTime	182.67	0.6431	0.7816	3.0598	3.0671	3.3128	4.2128	147.54
MaxTime	765.64	0.672	0.797	3.563	3.563	3.547	4.266	154.12
Problem P1. Node: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	2	26
MinVal	-	0.9738	0.9801	0.9801	0.9801	0.9738	0.9812	0.9893
MeanVal	-	0.9886	0.9889	0.989	0.989	0.9899	0.9911	0.9986
MaxVal	-	0.9966	0.9966	0.9966	0.9966	0.9966	1	1
MinTime	0.094	0	0	0.015	0.016	0	0.063	0.094
MeanTime	0.1528	0.0038	0.0097	0.0465	0.0463	0.0356	0.1297	0.249
MaxTime	0.359	0.016	0.016	0.063	0.063	0.047	0.172	0.281
Problem P1. Nodes: 100 (mesh). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	7
MinVal	-	0.99	0.99	0.99	0.99	0.9912	0.9928	0.9964
MeanVal	-	0.9951	0.9951	0.9953	0.9953	0.9954	0.9963	0.999
MaxVal	-	0.9974	0.998	0.9974	0.9974	0.9993	0.9995	1
MinTime	0.203	0.047	0.062	0.125	0.125	0.109	0.578	2.188
MeanTime	1.2009	0.0641	0.0719	0.2853	0.2965	0.27	0.65	5.4113
MaxTime	2.64	0.156	0.079	0.328	0.329	0.297	0.875	7.922
Problem P1. Nodes: 144 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	2
MinVal	-	0.9915	0.9915	0.9915	0.9915	0.9915	0.9934	0.9963
MeanVal	-	0.9949	0.995	0.9949	0.9949	0.9951	0.9965	0.9988
MaxVal	-	0.9982	0.9993	0.9982	0.9982	0.9982	0.9992	1
MinTime	0.829	0.25	0.297	0.5	0.5	0.5	1.703	14.063
MeanTime	3.1691	0.2799	0.3345	1.1516	1.1177	1.1407	1.924	36.538
MaxTime	9.75	0.375	0.453	1.671	1.657	1.594	2.547	41.719
Problem P1. Nodes: 196 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.992	0.992	0.992	0.992	0.9924	0.9934	0.9962
MeanVal	-	0.9945	0.9946	0.9945	0.9945	0.9946	0.9959	0.9983
MaxVal	-	0.9964	0.9973	0.9964	0.9964	0.9969	0.9988	0.9996
MinTime	4.5	0.781	0.968	1.546	1.531	1.516	3.219	113.89
MeanTime	19.084	0.7973	0.9858	3.7729	3.5577	3.535	5.5057	158.34
MaxTime	87.735	0.813	1.016	6	5.938	5.844	8.406	190.54

Table 9.12: Problem P1: Model, basic heuristics and TSH on random and mesh networks. Facilities: 25%.

	Model M3	HeurH3	H3Imp	ASH(R1)	ASH(R2)	ASH(R3)	ASH(R4)	ASH(R5)
Problem P3. Nodes: 50 (random). Paths: 250. Mean node number for paths: 4. Flows Range: 1-30.								
#OptVal	-	2	2	2	2	3	4	9
MinVal	-	0.9944	0.9944	0.9944	0.9944	0.9944	0.9961	0.998
MeanVal	-	0.9977	0.9977	0.9977	0.9977	0.998	0.9989	0.9998
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.14	0.031	0.015	0.031	0.031	0.031	0.046	0.046
MeanTime	0.1951	0.0345	0.0171	0.0361	0.0359	0.0358	0.0516	0.0904
MaxTime	0.328	0.047	0.032	0.047	0.047	0.047	0.063	0.156
Problem P3. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	2
MinVal	-	0.9955	0.9955	0.9955	0.9955	0.9955	0.9967	0.9977
MeanVal	-	0.9972	0.9972	0.9972	0.9972	0.9973	0.9984	0.9992
MaxVal	-	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	1
MinTime	0.484	0.563	0.281	0.578	0.578	0.578	0.625	1.203
MeanTime	1.2626	0.5735	0.2984	0.5859	0.5906	0.5861	0.7157	3.3953
MaxTime	6.25	0.593	0.313	0.594	0.594	0.594	0.844	5.61
Problem P3. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9947	0.9947	0.9947	0.9947	0.9947	0.9949	0.9981
MeanVal	-	0.9968	0.9968	0.9968	0.9968	0.9968	0.9972	0.9992
MaxVal	-	0.9984	0.9984	0.9984	0.9984	0.9984	0.9986	1
MinTime	1.875	4.062	1.546	4.203	4.203	4.203	4.688	49.984
MeanTime	2.4875	4.45	1.6266	4.5234	4.6437	4.4609	5.189	77.4156
MaxTime	5.156	6.047	2.281	5.843	6.265	5.125	6.125	131.437
Problem P3. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9959	0.9959	0.9959	0.9959	0.9959	0.9967	0.9984
MeanVal	-	0.9977	0.9977	0.9977	0.9977	0.9977	0.9983	0.9991
MaxVal	-	0.9989	0.9989	0.9989	0.9989	0.9989	0.9994	0.9997
MinTime	6.469	29.719	4.984	30.5	30.515	31.844	31.297	401.422
MeanTime	17.5376	29.7704	5.0046	30.9063	31.0936	31.8968	33.7547	557.57
MaxTime	31.188	29.813	5.031	34.265	31.953	31.953	35.86	699.313
Problem P3. Nodes: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	1	1	1	1	1	2	3
MinVal	-	0.9904	0.9904	0.9904	0.9904	0.9904	0.9953	0.9975
MeanVal	-	0.9961	0.9961	0.9961	0.9961	0.9961	0.9978	0.9989
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.14	0.031	0.015	0.031	0.031	0.031	0.047	0.062
MeanTime	0.1985	0.036	0.0156	0.0312	0.0359	0.0375	0.0533	0.1079
MaxTime	0.375	0.047	0.016	0.032	0.047	0.047	0.063	0.188
Problem P3. Nodes: 100 (mesh). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	1
MinVal	-	0.9936	0.9936	0.9941	0.9936	0.9936	0.9949	0.9972
MeanVal	-	0.9957	0.9957	0.9957	0.9957	0.9958	0.9967	0.9987
MaxVal	-	0.997	0.997	0.997	0.997	0.9982	0.9982	1
MinTime	0.688	0.578	0.297	0.593	0.594	0.578	0.672	3.813
MeanTime	1.025	0.5813	0.3078	0.6296	0.611	0.6014	0.742	5.814
MaxTime	2.641	0.594	0.313	0.891	0.625	0.61	0.875	7.578
Problem P3. Nodes: 144 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.996	0.996	0.996	0.996	0.9963	0.9971	0.9977
MeanVal	-	0.9975	0.9975	0.9975	0.9975	0.9976	0.9982	0.999
MaxVal	-	0.9985	0.9985	0.9985	0.9985	0.9985	0.9993	0.9999
MinTime	2.204	3.625	1.375	3.734	3.766	3.75	4.203	60.75
MeanTime	2.5238	4.1516	1.639	4.211	4.422	4.2547	5.6172	78.9109
MaxTime	3.516	5.375	2.047	5.578	5.594	5.547	7.5	109.063
Problem P3. Nodes: 196 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9954	0.9954	0.9954	0.9954	0.9955	0.9966	0.9975
MeanVal	-	0.9964	0.9964	0.9964	0.9964	0.9964	0.9976	0.9985
MaxVal	-	0.9969	0.9969	0.9969	0.9969	0.9969	0.9983	0.9996
MinTime	7.781	30.14	4.765	33.485	33.515	33.453	40.703	654.657
MeanTime	9.6219	32.8891	4.7765	35.986	35.5313	35.4625	45.7422	1010.35
MaxTime	15.766	37.063	4.796	39.703	39.766	40.422	51.078	1493

Table 9.13: Problem P3: Model, basic heuristics and ASH on random and mesh networks. Facilities: 50%.

	Model M3	HeurH3	H3Imp	TSH(R1)	TSH(R2)	TSH(R3)	TSH(R4)	TSH(R5)
Problem P3. Nodes: 50 (random). Paths: 250. Mean node number for paths: 4. Flows Range: 1-30.								
#OptVal	-	2	2	2	2	2	6	10
MinVal	-	0.9944	0.9944	0.9944	0.9953	0.9949	0.9961	1
MeanVal	-	0.9977	0.9977	0.9977	0.9978	0.9978	0.9992	1
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.14	0.031	0.015	0.047	0.062	0.062	0.234	0.531
MeanTime	0.1951	0.0345	0.0171	0.0717	0.0953	0.0673	0.2453	0.5378
MaxTime	0.328	0.047	0.032	0.078	0.11	0.079	0.266	0.547
Problem P3. Nodes: 100 (random). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	2
MinVal	-	0.9955	0.9955	0.9955	0.9955	0.9955	0.9971	0.9993
MeanVal	-	0.9972	0.9972	0.9973	0.9973	0.9973	0.9986	0.9997
MaxVal	-	0.9996	0.9996	0.9996	0.9996	0.9996	0.9998	1
MinTime	0.484	0.563	0.281	0.782	0.719	0.64	1.531	6.89
MeanTime	1.2626	0.5735	0.2984	0.8609	1.0172	0.8279	1.5406	12.5702
MaxTime	6.25	0.593	0.313	0.89	1.063	0.859	1.547	13.281
Problem P3. Nodes: 150 (random). Paths: 2250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	2
MinVal	-	0.9947	0.9947	0.9947	0.9947	0.9947	0.9952	0.9981
MeanVal	-	0.9968	0.9968	0.9968	0.9968	0.9968	0.9983	0.9996
MaxVal	-	0.9984	0.9984	0.9984	0.9984	0.9984	0.9999	1
MinTime	1.875	4.062	1.546	4.906	5.031	6.812	10.297	207.625
MeanTime	2.4875	4.45	1.6266	6.4516	7.5265	7.7359	11.3484	209.303
MaxTime	5.156	6.047	2.281	9.5	10.25	9.953	14.188	212.703
Problem P3. Nodes: 200 (random). Paths: 4000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9959	0.9959	0.9959	0.9959	0.9964	0.9975	0.9984
MeanVal	-	0.9977	0.9977	0.9977	0.9977	0.9978	0.9985	0.9994
MaxVal	-	0.9989	0.9989	0.9989	0.9989	0.9989	0.9994	0.9999
MinTime	6.469	29.719	4.984	34.281	34.5	35.406	60.265	847.344
MeanTime	17.5376	29.7704	5.0046	39.7593	41.0499	44.9157	60.3172	1449.15
MaxTime	31.188	29.813	5.031	48.578	47.031	46.094	60.531	1516.83
Problem P3. Nodes: 49 (mesh). Paths: 250. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	1	1	1	1	1	4	9
MinVal	-	0.9904	0.9904	0.9904	0.9904	0.9904	0.9954	0.9999
MeanVal	-	0.9961	0.9961	0.9961	0.9961	0.9964	0.9986	1
MaxVal	-	1	1	1	1	1	1	1
MinTime	0.14	0.031	0.015	0.031	0.047	0.062	0.234	0.485
MeanTime	0.1985	0.036	0.0156	0.0703	0.0938	0.0625	0.2374	0.4985
MaxTime	0.375	0.047	0.016	0.079	0.11	0.063	0.25	0.5
Problem P3. Nodes: 100 (mesh). Paths: 1000. Mean node number for paths: 5. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	4
MinVal	-	0.9936	0.9936	0.9941	0.994	0.9936	0.9949	0.9987
MeanVal	-	0.9957	0.9957	0.9957	0.9957	0.9957	0.9976	0.9997
MaxVal	-	0.997	0.997	0.9971	0.997	0.997	0.9995	1
MinTime	0.688	0.578	0.297	0.672	0.734	0.859	1.578	13.203
MeanTime	1.025	0.5813	0.3078	0.889	1.0078	0.8642	1.5829	13.3842
MaxTime	2.641	0.594	0.313	1.266	1.156	0.875	1.594	13.547
Problem P3. Nodes: 150 (mesh). Paths: 2250. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	1
MinVal	-	0.996	0.996	0.996	0.996	0.996	0.9971	0.9977
MeanVal	-	0.9975	0.9975	0.9975	0.9975	0.9976	0.9986	0.9994
MaxVal	-	0.9985	0.9985	0.9985	0.9985	0.9986	0.9995	1
MinTime	2.204	3.625	1.375	4.344	4.531	5.172	9.297	186.547
MeanTime	2.5238	4.1516	1.639	6.2842	6.6595	6.9579	10.775	190.81
MaxTime	3.516	5.375	2.047	8.812	9.781	8.844	13.75	194.75
Problem P3. Nodes: 200 (mesh). Paths: 4000. Mean node number for paths: 6. Flows Range: 1-30.								
#OptVal	-	0	0	0	0	0	0	0
MinVal	-	0.9954	0.9954	0.9954	0.9954	0.9954	0.9967	0.9979
MeanVal	-	0.9964	0.9964	0.9964	0.9964	0.9964	0.9978	0.9987
MaxVal	-	0.9969	0.9969	0.9969	0.9969	0.9969	0.9986	0.9996
MinTime	7.781	30.14	4.765	36.984	37.343	43.969	61.515	1414.8
MeanTime	9.6219	32.8891	4.7765	40.6546	46.1875	47.8077	62.95	1421.9
MaxTime	15.766	37.063	4.796	51.781	53.156	51.64	67.359	1449.11

Table 9.14: Problem P3: Model, basic heuristics and TSH on random and mesh networks. Facilities: 50%.

9.4 Sensitivity analysis and notations

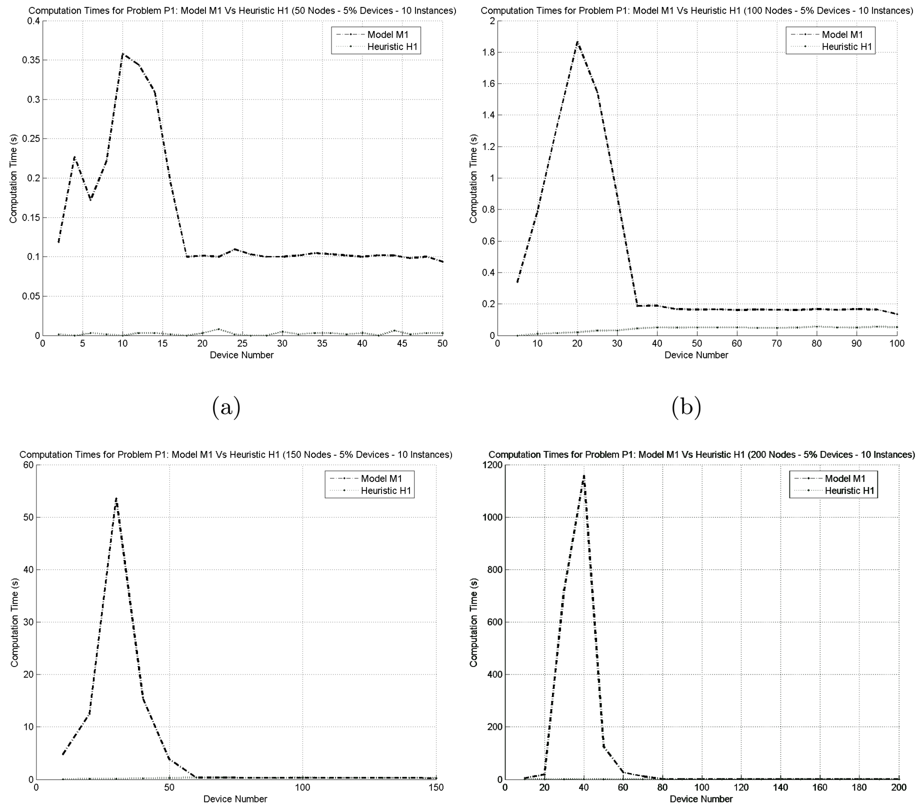
The four tables described in the previous section provide the results obtained for a prefixed value of the device number. In the following we provide a sensitivity analysis for *P1* and *P3* on the random networks to determine which parameters affect the quality of solution and/or computation times. Figure 9.11*(a-d)* represents the trend of the computation times of *M1* and basic heuristic *H1* over *10* instances for the random networks under investigation, depending on the number of devices. We see that the computation time of *M1* has a maximum for a certain value of the number of devices (m), around the *20%* of *N*. Moreover for large values of m, computation time is comparable with the one of the heuristic *H1*, which are not really affected by the number of devices. The same observation can also be made for the other heuristics proposed for problem *P1*, even if for large values of m the use of *ASH* and *TSH* may not be convenient.

Figures 9.12*(a-d)* represent the trends of computation times of *M3* and the heuristic *H3* and *Improved H3* over ten instances depending on the number of devices. Also in figure 9.12*(a-d)* the computation time for *M3* has a maximum for a certain value of the number of devices (m), between *10%* and *20%* of *N*. Beyond this point, computation time of *M3* gradually decreases and becomes very low for large values of m, whereas the heuristics presents the opposite trend. It is also important to note that for problem *P3*, with the exception of the case of the network with *50* nodes, we can identify a point, around *50%* of *N*, over which computation time of the basic heuristic *H3* (and obviously of *ASH* and *TSH*) are higher than those of the model. This situation does not occur for *Improved H3*, which returns solutions of the same quality of the basic heuristic *H3*, but its computation time is very close to that of *M3*.

Tables 9.15–9.18 show the results obtained for *P1* and *P3* varying number of paths for the network of *100* nodes, with m equal respectively to *5* and *10*.

Finally, in Table 9.19 we show the results obtained for *P1* and *P3* for the random network with *200* nodes,varying the number of iterations without improvement (δ) and varying the number of the total iterations in the *TSH* algorithm.

From tables 9.15–9.18 we can adfirm that the quality of solutions of models and methods is not affected by the number of paths considered for the network. Moreover, observing the results in 9.19, we can adfirm



(a) (b)

(c) (d)

Figure 9.11: Computation time: M1 vs. H1.

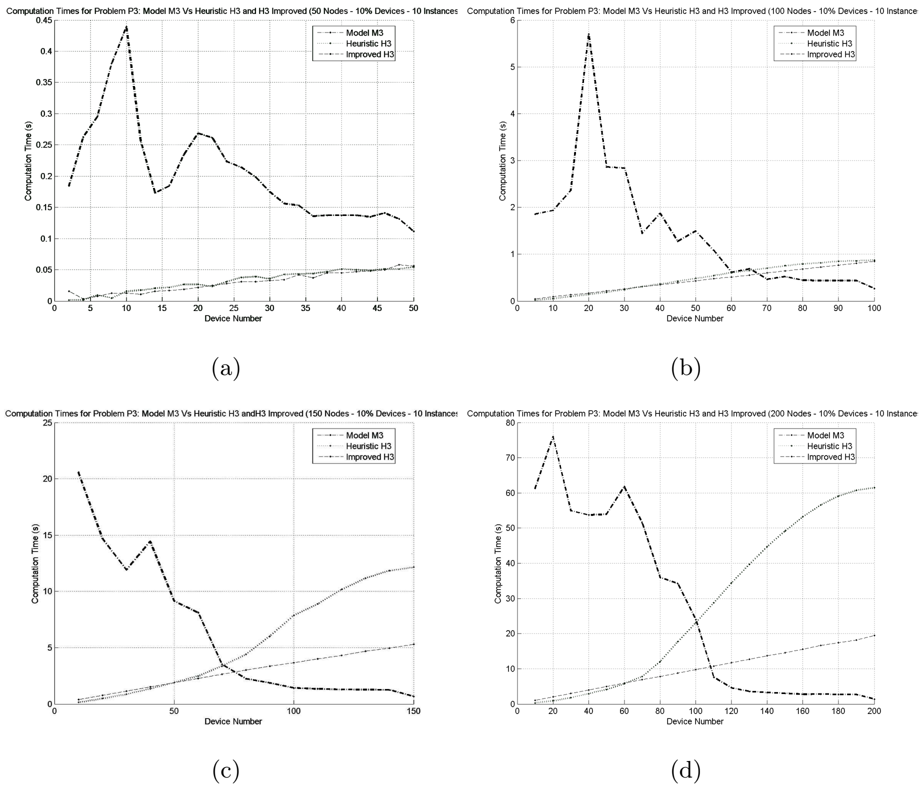


Figure 9.12: Computation time: M3 vs. H3 and Improved H3.

Problem P1. Nodes: 100. Devices: 5%. Flow Values: 1 - 30.									
Paths	Values	M1	H1	H1Imp	ASH(R1)	ASH(R2)	ASH(R3)	ASH(R4)	ASH(R5)
1000	# OptVal	-	35	35	35	35	36	35	44
	MinVal	-	0.990	0.990	0.990	0.990	0.991	0.990	0.991
	MeanVal	-	0.999	0.999	0.999	0.999	0.999	0.999	1.000
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	0.281	0	0	0.015	0	0.015	0.015	0.031
	MeanTime	0.336	0.015	0.014	0.024	0.020	0.024	0.022	0.042
	MaxTime	0.860	0.078	0.093	0.032	0.032	0.032	0.047	0.172
2000	# OptVal	-	48	48	48	48	49	48	50
	MinVal	-	0.995	0.995	0.995	0.995	0.995	0.995	1
	MeanVal	-	1	1	1.000	1.000	1.000	1.000	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	0.656	0.015	0.015	0.031	0.031	0.031	0.031	0.046
	MeanTime	0.724	0.024	0.023	0.042	0.040	0.041	0.044	0.061
	MaxTime	0.828	0.032	0.032	0.047	0.047	0.047	0.047	0.094
4000	# OptVal	-	24	24	24	24	29	24	50
	MinVal	-	0.978	0.978	0.978	0.978	0.978	0.978	1
	MeanVal	-	0.994	0.994	0.994	0.994	0.995	0.994	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	1.657	0.031	0.031	0.062	0.062	0.062	0.062	0.093
	MeanTime	2.164	0.039	0.047	0.072	0.071	0.074	0.078	0.152
	MaxTime	6.250	0.047	0.063	0.079	0.079	0.094	0.172	0.297
8000	# OptVal	-	11	11	11	11	19	11	50
	MinVal	-	0.990	0.990	0.990	0.990	0.990	0.990	1
	MeanVal	-	0.996	0.996	0.996	0.996	0.997	0.996	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	5.265	0.062	0.078	0.125	0.125	0.125	0.125	0.234
	MeanTime	5.924	0.072	0.096	0.139	0.133	0.138	0.142	0.344
	MaxTime	6.688	0.079	0.110	0.188	0.141	0.172	0.188	0.407

Table 9.15: Problem P1: ASH results and computation time depending on number of paths.

Problem P1. Nodes: 100. Devices: 5%. Flow Values: 1 - 30.									
Paths	Values	M1	H1	H1Imp	TSH(R1)	TSH(R2)	TSH(R3)	TSH(R4)	TSH(R5)
1000	# OptVal	-	35	35	36	36	43	35	50
	MinVal	-	0.990	0.990	0.990	0.990	0.994	0.990	1
	MeanVal	-	0.999	0.999	0.999	0.999	1.000	0.999	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	0.281	0	0	0.109	0.109	0.032	0.140	0.125
	MeanTime	0.336	0.015	0.014	0.117	0.119	0.101	0.152	0.415
	MaxTime	0.860	0.078	0.093	0.156	0.141	0.110	0.157	0.469
2000	# OptVal	-	48	48	48	48	49	48	50
	MinVal	-	0.995	0.995	0.995	0.995	0.995	0.995	1
	MeanVal	-	1	1	1.000	1.000	1.000	1.000	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	0.656	0.015	0.015	0.234	0.234	0.078	0.281	0.671
	MeanTime	0.724	0.024	0.023	0.239	0.242	0.218	0.286	0.728
	MaxTime	0.828	0.032	0.032	0.250	0.250	0.235	0.297	0.937
4000	# OptVal	-	24	24	43	43	36	24	50
	MinVal	-	0.978	0.978	0.982	0.996	0.984	0.978	1
	MeanVal	-	0.994	0.994	0.999	1.000	0.998	0.994	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	1.657	0.031	0.031	0.390	0.390	0.140	0.453	1.203
	MeanTime	2.164	0.039	0.047	0.406	0.404	0.379	0.458	1.400
	MaxTime	6.250	0.047	0.063	0.422	0.422	0.407	0.469	1.687
8000	# OptVal	-	11	11	11	11	36	11	50
	MinVal	-	0.990	0.990	0.990	0.990	0.990	0.990	1
	MeanVal	-	0.996	0.996	0.996	0.996	0.999	0.996	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	5.265	0.062	0.078	0.719	0.718	0.266	0.781	2.265
	MeanTime	5.924	0.072	0.096	0.777	0.733	0.686	0.846	2.941
	MaxTime	6.688	0.079	0.110	1.094	0.750	0.735	1.187	3.156

Table 9.16: Problem P1: TSH results and computation time depending on number of paths.

Problem P3. Nodes: 100. Devices: 10%. Flow Values: 1 - 30.										
Paths	Values	M3	H3	H3 Imp	ASH(R1)	ASH(R2)	ASH(R3)	ASH(R4)	ASH(R5)
1000	# OptVal	10	5	5	5	5	6	6	9
	MinVal	-	0.991	0.991	0.991	0.991	0.991	0.991	0.999
	MeanVal	-	0.997	0.997	0.997	0.997	0.998	0.998	1.000
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	1.875	0.046	0.047	0.046	0.046	0.047	0.991	0.093
	MeanTime	2.067	0.050	0.064	0.050	0.056	0.055	0.998	0.149
	MaxTime	2.468	0.078	0.171	0.063	0.063	0.063	1	0.297
2000	# OptVal	10	10	10	10	10	10	10	10
	MinVal	-	1	1	1	1	1	1	1
	MeanVal	-	1	1	1	1	1	1	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	7.562	0.078	0.093	0.078	0.093	0.078	1	0.172
	MeanTime	8.669	0.080	0.100	0.091	0.095	0.092	1	0.180
	MaxTime	10.406	0.093	0.110	0.110	0.110	0.109	1	0.188
4000	# OptVal	10	10	10	10	10	10	10	10
	MinVal	-	1	1	1	1	1	1	1
	MeanVal	-	1	1	1	1	1	1	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	39.281	0.140	0.203	0.218	0.234	0.218	1	0.328
	MeanTime	43.648	0.145	0.217	0.230	0.238	0.225	1	0.342
	MaxTime	48.031	0.156	0.219	0.250	0.250	0.234	1	0.360
8000	# OptVal	10	10	10	10	10	10	10	10
	MinVal	-	1	1	1	1	1	1	1
	MeanVal	-	1	1	1	1	1	1	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	229.157	0.265	0.438	0.421	0.437	0.421	1	0.672
	MeanTime	240.074	0.278	0.446	0.427	0.447	0.424	1	0.869
	MaxTime	258.031	0.328	0.453	0.438	0.454	0.438	1	0.906

Table 9.17: Problem P3: ASH results and computation time depending on number of paths.

Problem P3. Nodes: 100. Devices: 10%. Flow Values: 1 - 30.									
Paths	Values	M3	H3	H3Imp	TSH(R1)	TSH(R2)	TSH(R3)	TSH(R4)	TSH(R5)
1000	# OptVal	10	5	5	5	5	7	9	10
	MinVal	-	0.991	0.991	0.997	0.991	0.991	0.999	1
	MeanVal	-	0.997	0.997	1	0.997	0.998	1.000	1
	MaxVal	-	1	1	0.991	1	1	1	1
	MinTime	1.875	0.046	0.047	0.203	0.234	0.094	0.265	1.203
	MeanTime	2.067	0.050	0.064	0.203	0.241	0.170	0.266	1.223
	MaxTime	2.468	0.078	0.171	0.204	0.250	0.187	0.266	1.266
2000	# OptVal	10	10	10	10	10	10	10	10
	MinVal	-	1	1	1	1	1	1	1
	MeanVal	-	1	1	1	1	1	1	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	7.562	0.078	0.093	0.313	0.406	0.140	0.391	2.171
	MeanTime	8.669	0.080	0.100	0.327	0.411	0.275	0.403	2.192
	MaxTime	10.406	0.093	0.110	0.328	0.422	0.313	0.407	2.250
4000	# OptVal	10	10	10	10	10	10	10	10
	MinVal	-	1	1	1	1	1	1	1
	MeanVal	-	1	1	1	1	1	1	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	39.281	0.140	0.203	0.781	1.031	0.359	0.671	4.109
	MeanTime	43.648	0.145	0.217	0.792	1.033	0.649	0.717	4.128
	MaxTime	48.031	0.156	0.219	0.797	1.046	0.782	0.922	4.141
8000	# OptVal	10	10	10	10	10	10	10	10
	MinVal	-	1	1	1	1	1	1	1
	MeanVal	-	1	1	1	1	1	1	1
	MaxVal	-	1	1	1	1	1	1	1
	MinTime	229.157	0.265	0.438	1.484	1.953	0.703	1.656	8.015
	MeanTime	240.074	0.278	0.446	1.495	1.961	1.261	1.670	10.574
	MaxTime	258.031	0.328	0.453	1.500	1.969	1.484	1.672	10.891

Table 9.18: Problem P3: TSH results and computation time depending on number of paths.

Computation Times for network with 200 nodes varying settings of the TSH for P1 and P3											
	Obj Value	Rule: 1		Rule: 2		Rule: 3		Rule: 4		Rule: 5	
		5-20	5-20	5-20	5-20	5-20	5-20	5-20	5-20	5-20	5-20
	5% facilities	0,25	0,266	0,141	0,265	0,125	0,235	0,203	0,313	0,438	0,86
	z(heur): 9067										
P1	10% facilities	0,187	0,25	0,172	0,328	0,156	0,297	0,25	0,469	1,141	2,234
	z(heur): 12521										
	25% facilities	0,25	0,469	0,094	0,437	0,235	0,422	0,437	0,875	2,094	8,219
	z(heur): 15384										
	10% facilities	0,219	0,406	0,266	0,5	0,203	0,344	0,296	0,547	1,36	2,672
	z(heur): 43019										
P3	25% facilities	0,406	0,641	0,484	0,828	0,235	0,609	0,625	1,078	4,75	9,32
	z(heur): 59038										
	50% facilities	0,938	1,312	0,985	1,593	0,828	1,21	1,45	2,36	16,984	32,94
	z(heur): 68831										

Table 9.19: Problem P1 and P3: computation times and number of iterations for TSH.

that increasing the tuning parameters of *TSH*, the quality of solutions is the same, but computation times become very high.
These results are just a part of our experimental tests on the quality of solutions and on computation times for models and methods. We can summarize them as follows:

1. Quality of solutions is only affected by the configuration of paths on the network, whereas topology and dimension of the network, range of flow values and number of facilities, paths and iterations of *TSH* do not have significant effects on it.

2. Computation times are really affected by dimension of the network, configuration of paths and number of facilities, paths and iterations of *TSH*, but not by topology of the network or by the flow values.

Conclusions

In this thesis two City Logistics problems have been tackled: design of a two-echelon freight distribution system and location of infomobility devices on a network.

The design of a 2E-LRP is a strategical and tactical decisional problem and it has been modeled as a two-echelon location-routing problem (2E-LRP). It has been scarcely treated in literature and no exact neither heuristic solution methods have been proposed in literature for it.

In this thesis four models, differing for the kind of used variables, have been proposed for 2E-LRP, extending and/or adapting classical VRP formulations and a MDVRP formulation present in literature. Models have been experienced on test instances of varying dimensions with the usage of a commercial solver. Test instances have been generated through an original instance generator. The problem is NP-hard and the results obtained for the models, in terms of quality of solution and computation time, confirmed the need of approaching it with a heuristic method. To this aim a Tabu Search heuristic has been proposed and implemented. It is based on the decomposition of the whole problem in four subproblems, one FLP and one VRP for each echelon. The four sub-problems are sequentially and iteratively solved and their solutions are opportunely combined in order to determine a good global solution. Tabu Search has been experienced on three set of small, medium and large instances and the obtained results have been compared with the results of the models. Experimental results prove that the proposed TS is effective in terms of quality of solutions and computation times in the most of the solved instances. The proposed Tabu Search presents a modular structure which makes it very flexible. Therefore it could be easily integrated with intensification criteria, extended with other constraints (such as maximum length of the routes, more fleets of vehicles for each echelon) and adapted to the asymmetric case.

The problem problem is new and the research field is unexplored. Therefore future research work should move towards exact approaches in order to test the heuristics, which seem to be anyway the more effective way to approach this problem.

The second theme of this thesis is related to infomobility service location. Four problems of flow intercepting facility location have been treated with several models and methods, some of them present in literature and other proposed by us. An extensive computational experimentation of them has been carried out in terms of quality of solutions and computation times. We can adfirm that heuristics return very good solutions, very close to the optimum, even if in some cases (for networks with less than *200* nodes and for large values of facilities to locate) they require computation times not far from those required by the mathematical models. We have successfully applied FIFLP models to transportation and communication networks. Possible extensions of this work should concern the development of exact or heuristic methods for the specializations and integrations of the proposed models and for the mobile facility location case, which has been modeled in the thesis.

We think that the two approached problems could be merged in order to reduce the impact of private transportation in congested urban areas. The idea is to prevent the penetration of a large number of private vehicles in the city center intercepting them along their pre-planned trip and providing then an efficient public transportation system. In practice users traveling from the city outskirts to the city center can be invited or forced to park their vehicles in parking areas and then use public transportation system to reach their final destinations. To solve this problem new flow-interception location-routing model could be built to find a good location of *urban multimodal platforms* to intercept as much vehicular flow as possible and to define the best routes for the public transportation lines starting from the platforms.

Appendix

In order to evaluate the goodness of the models and methods presented in the thesis, several instances have been generated through an original *instance generator*, developed in C++.

The instances have been generated with the aim of representing a possible structure of a freight distribution system for an urban area. Customers and facilities are all located within a circular area. Customers are located around the center, whereas secondary and primary facilities are located at increasing distances from it. The instance generation is based on three steps:

- *Step 1-Definition of the investigation area dimensions*: we have to define the dimensions of the areas where customers, satellites and platforms will be located. We define three round concentric areas, referred as *Area 1*, *Area 2* and *Area 3* of increasing dimension (figure 9.13).

  The input for the instance generator is the ray value of each area, which will be referred as ray_1, ray_2, and ray_3 respectively for *Area 1*), *Area 2* and *Area 3*. Obviously the following relation has to be satisfied: $ray_1 \leq ray_2 \leq ray_3$.

- *Step2-Definition of the instance*. We provide to the instance generator the number of customers, satellites and platforms to locate. These values will be referred respectively as Z, S and P.

- *Step3-Definition of spatial distribution*. We have to define how to distribute customers and facilities (satellites and platforms) in the different areas. The following criteria have been used:

  Customer distribution: customers are all randomly distributed within *Area 1*.

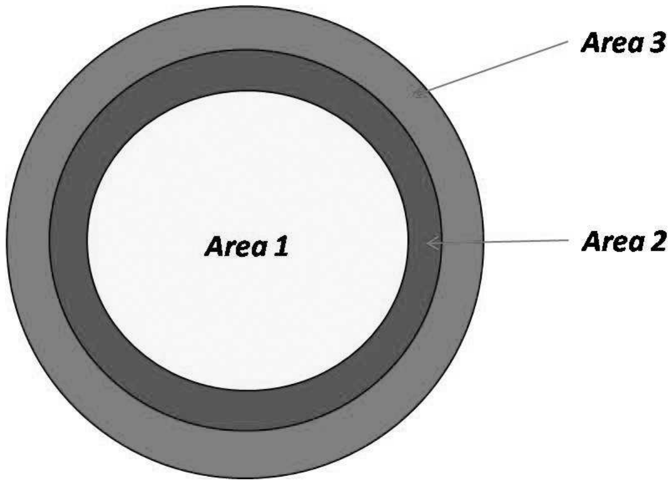


Figure 9.13: Instance structure

Satellite distribution: a satellite can be located within *Area 1* and *Area 2*. The number of satellites to locate in each area is a parameter of the instance generator. More precisely a percentage α is defined and satellites will be distributed as follows:
- $\alpha\%$ of the total number of satellites is located in *Area 2*;
- $(1 - \alpha\%)$ of the total number of satellites is located within *Area 1*.
In each area the satellites are randomly distributed.

Platform distribution: the same criterion for the distribution of the satellites is used for the platform, but for *Area 2* and *Area 3*. Therefore, also in this case, the number of platforms to locate in each area is chosen at the beginning as a percentage of the total platform locations:
- $\alpha\%$ of the total number of platforms is located in *Area 3*;
- $(1 - \alpha\%)$ of the total number of platfomrs is located in *Area 2*.

In each area the satellites are randomly distributed.

The models and methods present in the thesis have been experienced on three set of instances which differ for the distribution of the satellites:

1. *Test set I1*: satellites are all located in *Area 2*;
2. *Test set I2*: half satellites are located in *Area 1* and the others in *Area2*;
3. *Test set I3*: satellites are all located in *Area 1*.

For each test set, different combinations of customers, satellites and platforms have been used:

- $Z = \{8, 9, 10, 12, 15, 20, 25, 50, 75, 100, 150, 200\}$;
- $S = \{3, 4, 5, 8, 10, 15, 20\}$;
- $P = \{2, 3, 4, 5\}$

Customer demands have been randomly generated in the range $[1, 100]$. Facility and vehicle capacity values vary with the size of the instances:

- satellite capacity (K_s): for instances up to 100 customers, their capacity is randomly generated in the range $[300, 600]$, whereas for instances with 150 and 200 customers their capacity is randomly generated in the range $[500, 1500]$. In both cases location costs are a linear function of the capacity values and vary in the range $[40, 80]$.
- platform capacity (K_p): for instances up to 100 customers, their capacity is randomly generated in the range $[1000, 2000]$, whereas for instances with 150 and 200 customers their capacity is randomly generated in the range $[3000, 6000]$. Also in this case, the location costs have been determined as a linear function of the capacity values and vary in the range $[150, 250]$.
- urban trucks capacity (UG): for instances up to 10 customers, their capacity is equal to *500*; for instances up to 100 customers their capacity is equal to *1500*; for instances up to 200 customers, their capacity is equal to *3000*. No cost is considered for the usage of a vehicle.
- city freighters capacity (UV): for instances up to 10 customers, their capacity is equal to *100*; for instances up to 100 customers their capacity is equal to *200*; for instances up to 200 customers,

their capacity is equal to *500*. No cost is considered for the usage of a vehicle.

In the thesis the instances will be indicated with the following notation: $Testset - PSZ$. For example *I1-51050* is an instance of set I with 5 platforms, 10 satellites and 50 customers.

List of Figures

2.1 Two-echelon distribution system. 28

3.1 Two-echelon location routing problem representation. . . 57

5.1 Tabu Search scheme. 88
5.2 First feasible solution. 91
5.3 Combining sub-problems on a single-echelon. 105

7.1 Paths on a network and related path-node incidence matrix $\boldsymbol{B}$. 138
7.2 Four possible shapes of the a_{pi} coefficients. 142

9.1 Problem P1: Model M1 solutions Vs Heuristic method solutions. 163
9.2 Problem P1: Model M1 solutions Vs ASH solutions (five rules). 164
9.3 Problem P1: Model M1 solutions Vs TSH solutions (five rules). 164
9.4 Problem P2: Model M2 solution vs Heuristic H2 solutions over a single instance. 165
9.5 Problem P2: Model M2 solution vs. Heuristic H2 solutions over 50 instances. 166
9.6 Problem P3: Model M3 solutions vs. Heuristic method solutions. 167
9.7 Problem P3: M3 solutions vs. ASH solutions (five rules). 167
9.8 Problem P1: M3 solutions vs. TSH solutions (five rules). 168
9.9 Problem P4: Model M4 solution vs. Heuristic H4 solutions over a single instance. 169
9.10 Problem P4: Model M4 solution vs. Heuristic H4 solutions over 50 instances. 169

9.11 Computation time: M1 vs. H1. 184
9.12 Computation time: M3 vs. H3 and Improved H3. 185
9.13 Instance structure 192

List of Tables

4.1 Results of 3-index and ab-based formulations on small instances I1. 78
4.2 Results of 3-index and ab-based formulations on small instances I2. 79
4.3 Results of 3-index and ab-based formulations on small instances I3. 79
4.4 Results of decomposition approach on large-medium instances I1. 80
4.5 Results of decomposition approach on large-medium instances I2. 80
4.6 Results of decomposition approach on large-medium instances I3. 81

6.1 Tabu Search criteria. 110
6.2 Tabu Search parameters. 111
6.3 Tabu Search setttings 1 and 2. 112
6.4 Tabu Search settings 3 and 4. 113
6.5 Experimental results of TS settings on test instances I1. . 114
6.6 Experimental results of TS settings on test instances I2. . 115
6.7 Experimental results of TS settings on test instances I3. . 116
6.8 Results of diversification for TS setting 1. 117
6.9 Results of diversification for TS setting 4. 118
6.10 Tabu Search vs. models on small instances I1. 120
6.11 Tabu Search vs. models on small instances I2. 120
6.12 Tabu Search vs. models on small instances I3. 121
6.13 Tabu Search vs. decomposition approach on medium-large instances I1. 121
6.14 Tabu Search vs. decomposition approach on medium-large instances I2. 122

6.15 Tabu Search vs. decomposition approach on medium-large instances I3. 122

9.1 Test Network and Parameter Setting. 162
9.2 Test parameters. 170
9.3 Problem P1: Model, basic heuristics and ASH on random and mesh networks. Facilities: 5%. 171
9.4 Problem P1: Model, basic heuristics and TSH on random and mesh networks. Facilities: 5%. 172
9.5 Problem P3: Model, basic heuristics and ASH on random and mesh networks. Facilities: 10%. 173
9.6 Problem P3: Model, basic heuristics and TSH on random and mesh networks. Facilities: 10%. 174
9.7 Problem P1: Model, basic heuristics and ASH on random and mesh networks. Facilities: 10%. 175
9.8 Problem P1: Model, basic heuristics and TSH on random and mesh networks. Facilities: 10%. 176
9.9 Problem P3: Model, basic heuristics and ASH on random and mesh networks. Facilities: 25%. 177
9.10 Problem P3: Model, basic heuristics and TSH on random and mesh networks. Facilities: 25%. 178
9.11 Problem P1: Model, basic heuristics and ASH on random and mesh networks. Facilities: 25%. 179
9.12 Problem P1: Model, basic heuristics and TSH on random and mesh networks. Facilities: 25%. 180
9.13 Problem P3: Model, basic heuristics and ASH on random and mesh networks. Facilities: 50%. 181
9.14 Problem P3: Model, basic heuristics and TSH on random and mesh networks. Facilities: 50%. 182
9.15 Problem P1: ASH results and computation time depending on number of paths. 186
9.16 Problem P1: TSH results and computation time depending on number of paths. 186
9.17 Problem P3: ASH results and computation time depending on number of paths. 187
9.18 Problem P3: TSH results and computation time depending on number of paths. 187
9.19 Problem P1 and P3: computation times and number of iterations for TSH. 188

Bibliography

[1] ADLER, J.L.: Investigating the learning effects of route guidance and traffic advisories on route choice behaviour. Transportation Research Part C, 9, (2001).

[2] AMBROSINO, D., SCUTELLÁ, M.G.: Distribution network design: New problems and related models. European Journal of Operational Research, 165, 610624, (2005).

[3] ALBAREDA-SAMBOLA, M., DIAZ, J., FERNÁNDEZ, E.: A compact model and tight bounds for a combined location-routing problem. Computers and Operations Research, 32 , 407-428, (2005).

[4] AVERBAKH, I., BERMAN, O.: Locating flow-capturing units on a network with multi-counting and diminishing returns to scale. European Journal of Operational Research, 91, 3,495-506, (1996).

[5] BALAKRISHNAN, A., WARD, J.E., WONG, R.T.: Integrated facility location and vehicle routing models: Recent work and future prospects. American Journal of Mathematical and Management Sciences, 7, 35-61, (1987).

[6] BARRETO, S., FERRIERA, C., PAIXAO, J., SANTON, B.S.: Using clustering analysis in a capacitated location-routing problem, European Journal of Operational Research, 179, 3, 968-977, (2007).

[7] BEN-AKIVA, M., BIERLAIRE, M., BURTON, D., KOUTSOPOULOS, H., MISHALANI, R.: Network State Estimation and Prediction for Real-Time Traffic Management, Network and Spatial Economics, 1, 3-4, 293-318,(2001).

[8] BERMAN, O.: The maximizing market size discretionary facility location problem with congestion. Socio-Economic Planning Science, 29, 1, 39-46,(1995).

[9] BERMAN, O.: Deterministic flow-demand location problems. The Journal of the Operational Research Society, 48, 1, 75-81,(1997).

[10] BERMAN, O., BERTSIMAS, D., LARSON, C.R.: Locating Discretionary Service Facilities, II: Maximizing Market Size, Minimizing Inconvenience. Operations Research, 43, 4, 623-632, (1995).

[11] BERMAN, O., KRASS, D.: Flow intercepting spatial interaction model: a new approach to optimal location of competitive facilities. Location Science, 6 , 1-4, 41-65, (1998).

[12] BERMAN, O., KRASS, D., XU, C.W.: Locating Flow-Intercepting Facilities: New Approaches and Results. Annals of Operations Research, 60, 1, 121-143, (1995).

[13] BERMAN, O., LARSON, R.C., FOUSKA, N.: Optimal Locating of Discretionary Facilities. Transportation Science, 26, 3, 201-211, (1992).

[14] BESTUFS. Best Urban Freight Solutions. European Union project, http://www.bestufs.net.

[15] BIANCO, L., CONFESSORE, G., REVERBERI, P.: A network based model for traffic sensor location with implications on O/D matrix estimates. Transportation Science, 35, 1, 50-60, (2001).

[16] BOCCIA, M., SFORZA, A., STERLE, C.: Flow Intercepting Facility Location: Problems, Models and Heuristics. JMMA, Journal of Mathematical Modelling and Algorithms, 8, 1, 35-79, (2009).

[17] BOYLE, L.N., MANNERING, F.: Impact of traveler advisory systems on driving speed: some new evicence. Transportation Research Part C, 12, (2004).

[18] BROWNE, M., ALLEN, S., ANDERSEN, S., WOODBURN, A.: Urban Freight Consolidation Centres. In Taniguchi, E. and Thompson, R.G., editors, Recent Advances in City Logistics, 253265, Elsevier, Amsterdam, (2006).

[19] BRUNS, A., KLOSE, A.: A location first-route second heuristic for a combined location-routeing problem. In U. Zimmermann, U. Derigs, W. Gaul, R. Mohring, et K. Schuster (Eds.), Operations Research Proceedings. Springer, (1996).

[20] CANTIENI, R.G., IANNACCONE, G., BARAKAT, C., DIOT, C., THIRAN, P.. Reformulating the Monitor Placement Problem: Optimal Network-Wide Sampling. Information Sciences and Systems, 2006.

[21] CASCETTA, E., NGUYEN, S.: A unified framework for estimating or updating Origin/Destination trip matrices from traffic counts. Transportation Research B, 22, 437-455, (1988).

[22] CHAUDET, C., FLEURY, E., LASSOUS, G.I.: Optimal positioning of active and passive monitorings. Research Report 5273, INRIA, 2004.

[23] CHAUDET, C., FLEURY, E., RIVANO, H., LASSOUS, G.I., VOGE, E.M.: Optimal positioning of active and passive monitoring devices. Proceedings of the 2005 ACM conference on Emerging network experiment and technology, 71-82, (2005).

[24] CHEN, C.H., TING, C.J. : A hybrid lagrangian heuristic/simulated annealing algorithm for the multi-depot location routing problem. Proceedings of the Eastern Asia Society for Transportation Studies, 6, (2007).

[25] CHIEN, T.W.: Operational estimators for the length of a traveling salesman tour. Computers and Operations Research, 19, 469-478, (1992).

[26] CHIEN, T.W.: Heuristic Procedures for Practical-Sized Uncapacitated Location-Capacitated Routing Problems. Decision Sciences, 24, 5, 993-1021, (1993).

[27] CHRISTOPHIDES, N., EILON, S.: Expected distances in distribution problems. Operational Research Quarterly, 20, 437-443.

[28] CHRISTOPHIDES, N., MINGOZZI, A., TOTH, P.: The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (Eds.), Combinatorial Optimization. Wiley, Chichester, 315338, (1979).

[29] CHURCH R., ReVELLE, C.: The maximal covering location problem. Papers in Regional Science Association, 32, 1, 101-118, (1974).

[30] CITY PORTS. A Network of Cities Following a Co-ordinated Approach to Develop Feasible and Sustainable City Logistics Solutions. European Union project, http://www.cityports.net.

[31] CIVITAS Initiative. City-Vitality-Sustainability. European Union project, http://www.civitas-initiative.org.

[32] CLARKE, G., WRIGHT, J.: Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12, 4, 568-581, (1964).

[33] CORNUEJOLS, G., FISHER, M.L., NEMHEUSER, G.L.: Location of bank accounts to optimize float: An analysis study of exact and approximate algorithms, Management Science, 23, 8, 789-810, (1977).

[34] CRAINIC T.G., RICCIARDI, N., STORCHI, G.: Advanced freight transportation systems for congested urban areas. Transportation Research Part C: Emerging Technologies, 12, 2, 119-137, (2004).

[35] CRAINIC T.G., RICCIARDI, N., STORCHI, G.: Planning models for city logistics Operation. Techinical report, (2007).

[36] CRAINIC T.G., MANCINI, S., PERBOLI, G., TADEI, R.: Clustering-based heuristic for the two-echelon vehicle routing problem. CIRRELT-2008-46 november report, (2008).

[37] DONDO, R., CERDÁ, J.: A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. European Journal of Operational Research, 176, 14781507, (2007).

[38] DRÉO, P., PÉTROWSKI A., SIARRY, P., TAILLARD, E.: Metaheuristics for hard optimization: methods and case studies. Springer Berlin, (2003).

[39] DUIN, J.V.: Evaluation and Evolution of the City Distribution Concept. In Urban Transport and the Environment for the 21st Century III, 327337. WIT Press, Southampton, (1997).

[40] EILON, S., WATSON-GANDY, C.D.T., HEILBRON, A.: A vehicle fleet costs more. International Journal of Physical Distribution & Logistics Management, 1, 3, 126-132, (1971).

[41] EWGLA: http://www.vub.ac.be/EWGLA/literature.html, (2003).

[42] FISCHETTI, M., SALAZAR, G.J.J, TOTH, P.: The symmetric generalized travelling salesman problem polytope. Networks, 26, 113-123, (1995).

[43] GENDREAU, M., LAPORTE, G., PARENT, I.: Heuristics for the Location of Inspection Station on a Network. Naval Research Logistic, 47, 4, 287-303, (2000).

[44] GENTILI, M., MIRCHANDANI, P.B.: Locating Active Sensors on Traffic Networks. Annals of Operations Research, 35, 1, 229-257, (2005).

[45] GLOVER, F.: Future paths for integer programming and links to artificial intelligence. Computers and Operations Research, 5, 513-549, (1986).

[46] GLOVER, F.: Tabu search. part I. ORSA Journal on Computing, 1, 190-206, (1989).

[47] GLOVER, F.: Tabu search. part II. ORSA Journal on Computing, 2, 4-32, (1990).

[48] GLOVER, F., LAGUNA, M.: Tabu Search. Boston, USA: Kluwer Academic Publishers, (1997).

[49] GRAGNANI, S., VALENTI, G., VALENTINI, M.: City Logistics in Italy: A National Project. In Taniguchi, E. and Thompson, R.G., editors, Logistics Systems for Sustainable Cities, 279293, Elsevier, Amsterdam, (2004).

[50] HAKIMI, S.L.: Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph. Operations Research, 12, 450-459, (1964).

[51] HAKIMI, S.L.: Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems. Operations Research, 13, 462-475, (1965).

[52] HANSEN, P.H., HEGEDAHL, B., HJORTKJAER, S., OBEL, B.: A heuristic solution to the warehouse locationrouting problem, European Journal of Operational Research, 76, 111127, (1994).

[53] HODGSON, M.J.: The location of Public Facilities Intermediate to the Journey to Work. European Journal of Operational Research 6, 2, 199-204, (1981).

[54] HODGSON, M.J., ROSING, K.E., ZHANG, J.: Locating vehicle inspection stations to protect a transportation network. Location Science, 5, 3, 198, (1997).

[55] HU, C., LIU, B., LIU, Z., GAO, S., WU, D.: Optimal Deployment of Distributed Passive Measurement Monitors. International Conference on Communication, ICC'06, IEEE, 2, 621-626, (2006).

[56] HUYNH, N., CHIU, Y.C., MAHMASSANI, H.S.: Finding Near-Optimal Locations for Variable Message Signs for Real-Time Network Traffic Management. Journal of Transportation Research Board, 1856, 34-53, (2003).

[57] INSTITUTE OF CITY LOGISTICS: http://ww.citylogistics.org.

[58] JACOBSEN, S.K., MADSEN, O.B.G.: A comparative study of heuristics for a two-level locationrouting problem. European Journal of Operational Research, 5, 378-387, (1980).

[59] KHULLER, S., MOSS, A., NAOR, J.S.: The budgeted maximum coverage problem. Information Processing Letters, 70, 1, 39-45, (1999).

[60] KARP, R.:Reducibility among combinatorial problems. In Miller, R. and Thatcher, J. (eds.), Complexity of Computer Computations. Plenum Press, New York, 85-104, (1972).

[61] LAPORTE, G.: Location-routing problems. In B.L. Golden & A.A. Assad (Eds), Vehicle routing: Methods and studies. Amsterdam: North-Holland, (1988).

[62] LAPORTE, G.: A survey of algorithms for location-routing problems. Investigación Operativa, 1, 93123, (1989).

[63] LAPORTE, G., GENDREAU, M., POTVIN, J.Y., SEMET, F.: Classical and modern heuristics for the vehicle routing problem. International Transactions in Operational Research, 7, 285300, (2000).

[64] LAPORTE, G., NOBERT, Y.: An exact algorithm for minimizing routing and operating costs in depot location. European Journal of Operational Research 6, 224-226, (1981).

[65] LAPORTE, G., NOBERT, Y., ARPIN, D.: An exact algorithm for solving a capacitated location-routing problem. Annals of Operations Research 6, 293-310, (1986).

[66] LAPORTE, G., NOBERT, Y., PELLETIER, P.: Hamiltonian location problems. European Journal of Operational Research 12, 82 89, (1983).

[67] LAPORTE, G., NOBERT, Y., TAILLEFER, S.,: Solving a family of multi-depot vehicle routing and location-routing problems. Transportation Science 22, 161172, (1988).

[68] LENSTRA, J. K., RINNOY KAN, A.H.G.: Complexity of vehicle routing and scheduling problems, Networks, 11, 2, 221-227, (1981).

[69] LIN, S.: Computer solutions of the TSP. Bell Systems Technical Journal 44, 2245-2269, (1965).

[70] LIN, C.K.Y., CHOW, C.K., CHEN, A.: A location-routingloading problem for bill delivery services. Computers and Industrial Engineering 43, 525, (2002).

[71] LIN, S. KERINIGHAN, B.: An effective heuristic algorithm for the traveling salesman problem. Operations Research, 21 , 498-516, (1973).

[72] MADSEN, O.B.G.: Methods for solving combined two level location-routing problems of realistic dimensions. European Journal of Operational Research, 12, 3, 295-301, (1983).

[73] MAGGI, E.: La logistica urbana delle merci: Aspetti economici e normativi. Polipress (Eds.), (2007).

[74] MARANZANA, F.E.: On the location of supply points to minimise transport costs. Operational Research Quarterly 15, 261270, (1964).

[75] MELECHOVSKÝ, J., PRINS, C., CALVO, R.W.: A metaheuristic to solve a location-routing problem with non-linear costs. Journal of Heuristics, 11, 375-391, (2005).

[76] MILLER, C.E., TUCKER, A.W., ZEMLIN, R.A.: Integer Programming Formulation of Traveling Salesman Problems. Journal of the ACM (JACM) archive, 7, 4, 326-329, (1960).

[77] MIN, H., JAYARAMAN, V., SRIVASTAVA, R.: Combined location-routing problems: A synthesis and future research directions. European Journal of Operational Research, 108, 1-15, (1998).

[78] MIRCHANDANI, P.B., REBELLO, R., AGNETIS, A.: The inspection station location problem in hazardous materials transportation: Some heuristics and bounds. INFOR, 33, 2, 100-113, (1995).

[79] NAGY, G., SALHI, S.: Nested Heuristic Methods for the Location-Routeing Problem. The Journal of the Operational Research Society, 47, 9, 1166-1174, 1996.

[80] NAGY, G., SALHI, S.: Location-routing: Issues, models and methods. European Journal of Operational Research, 177, 649672, (2007).

[81] OECD. Delivering the Goods: 21st Century Challenges to Urban Goods Transport. Technical report, Organisation for Economic Co-operation and Development, OECD Publishing, http://www.oecdbookshop.org, (2003).

[82] OR, I., PIERSKALLA W.P.: A transportation location-allocation model for regional blood banking. AIIE Transactions, 11, 86-95, (1979).

[83] ÖZYURT, Z., AKSEN, D.: Solving the Multi-Depot Location-Routing Problem with Lagrangian Relaxation. In Book Series Operations Research/Computer Science Interfaces, 37, 125-144, (2007)

[84] PERL, J., DASKIN, M.S.: A warehouse location-routing problem. Transportation Research Part B, 19, 5, 381-396, (1985).

[85] PRINS, C., PRODHON, C., WOLFER-CALVO, R.: Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR, 4, 221-238, (2006).

[86] ROSENKRANTZ, D.J., TAYI, G.K., RAVI, S.S.: Algorithms for Path-Based Placement of Inspection Stations on Networks. INFORMS Journal on Computing, 12, 2, 136-149, (2000).

[87] RUSS0, F., COMI, A.: A state of the art on urban freight distribution at European scale. ECOMM 2004, Lion, France, European Conference on Mobility Management, //http:www.epomm.org/, (2004).

[88] SALHI, S., RAND, G.: Theory and methodology. The effect of ignoring routes when locating depots. European journal of operational research, 39, 2, 150-156, 1989.

[89] SRIVASTAVA, R.: Alternate solution procedures for the location-routing problem. Omega International Journal of Management Science 21, 4, 497-506, (1993).

[90] SRIVASTAVA, R., BENTON, W.C.: The location-routing problem: Considerations in physical distribution system. Computers and Operations Research 17, 5, 427-435, (1990).

[91] SUH, K., GUO, Y., KUROSE, J., TOWSLEY, D.: Locating network monitors: complexity, heuristics, and coverage. Computer Communications, 29, 10, 1564-1577, (2006).

[92] TANIGUCHI, E., HEIJDEN, R.V.D.: An Evaluation Methodology for City Logistics. Transport Reviews, 20, 1, 6590, (2000).

[93] TANIGUCHI, E., THOMPSON, R., YAMADA, T., DUIN, J.V.: City Logistics: Network Modelling and Intelligent Transport Systems. Pergamon, Amsterdam, (2001a).

[94] THOMPSON, R., TANIGUCHI, E.: City Logistics and Transportation. In Handbook of Logistics and Supply-Chain Management, 393405, Elsevier, Amsterdam, (2001).

[95] TOMAS, A.P.. Solving Optimal of traffic counting posts at urban intersections in CLPMICAI 2002. Advances in Artificial Intelligence, 2313, 247-266, (2002).

[96] TOTH, P., VIGO, D.: The Vehicle Routing Problem. SIAM, Philadelphia, (2002).

[97] TOTH, P., VIGO, D.: Models, relaxations and exact approaches for the capacitated vehicle routing problem. Discrete Applied Mathematics, 123, 487512, (2002).

[98] TRENDESETTER. Setting Trends for Sustainable Urban Mobility. European Union project, http://www.trendsetter-europe.org.org.

[99] TUZUN, D., BURKE, L.I.: A two-phase tabu search approach to the location routing problem, European Journal of Operational Research, 116, 8799, (1999).

[100] UNIONTRASPORTI: City Logistics: strategie d'intervento per il rifornimento delle reti commerciali al dettaglio. Proposta di Transit Point urbani per la provincia di Napoli, (2009).

[101] WANG, X., SUN, X., FANG,Y.: A two-phase hybrid heuristic search approach to the location-routing problem. IEEE International Conference on Systems, Man and Cybernetics, 4, 3338-3343, (2005).

[102] WEBB, M.H.J.: Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points. Operation Research Quartely, 23, 3, 333-344, (1972).

[103] Workshop Infomobilitá e Telematica. Napoli, 2004.

[104] WOTTON, J.R., GARCÁ-ORTIZ, A.: Intelligent Transportation Systems and Intelligent Transportation Systems. Geoinformatica, 4, 2, 2000.

[105] WREN, A., HOLLIDAY, A.: Cost Functions in the Location of Depots for Multiple-Delivery Journeys. Operation Research, 19, 3, 311-320, (1968).

[106] WU, T.H., CHINYAO, L., BAI, J.W.: Heuristic solutions to multi-depot location-routing problems. Computers & Operations Research, 29, 1393-1415, (2002).

[107] YANG, C., CHOOTINAN, P., CHEN, A.: Traffic Counting Location Planning using genetic algorithm. Journal of the Eastern Asia Society for Transportation Studies, 5, 898-913, (2003).

[108] YANG, H., YANG, C., LIPING, G.: Models and algorithms for the screen line-based traffic-counting location problems. Computers and Operations Research, 33, 3, 836-858, (2006).

[109] YANG, H., ZHOU, J.: Optimal Traffic Counting location for Origin-Destination matrix estimation. Transportation Research B, 32, 2, 109-126, (1998).