Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado

Models, Algorithms, and Heuristics for Multiobjective Commercial Territory Design

Tesis presentada por

María Angélica Salazar Aguilar

En opción al grado de

Doctor en Ingeniería con Especialidad en Ingeniería de Sistemas

Ciudad Universitaria, Abril de 2010

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica
División de Estudios de Posgrado

Models, Algorithms, and Heuristics for Multiobjective Commercial Territory Design

Tesis presentada por

María Angélica Salazar Aguilar

En opción al grado de

Doctor en Ingeniería con Especialidad en Ingeniería de Sistemas

Ciudad Universitaria, Abril de 2010

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis "MODELS, ALGORITHMS, AND HEURISTICS FOR MULTIOBJECTIVE COMMERCIAL TERRITORY DESIGN", realizada por la alumna MARÍA ANGÉLICA SALAZAR AGUILAR, matrícula 1294370, sea aceptada para su defensa como opción al grado de Doctor en Ingeniería con Especialidad en Ingeniería de Sistemas.

Comité de Tesis

Dr. Roger Z. Ríos Mercado
Asesor

Dr. José Luis González Velarde
Asesor

Dr. Mauricio Cabrera Ríos
Revisor

Dra. Yasmín A. Ríos Solis
Revisor

Dr. Rafael Caballero Fernández
Revisor

Vo. Bo.

Dr. Moisés Hinojosa Rivera
Subdirector
División de Estudios de Posgrado

CIUDAD UNIVERSITARIA, ABRIL DE 2010

VITA

Angélica was born in Querétaro, Querétaro, México on February 16, 1981, the eldest daughter of Consuelo Aguilar and J. Guadalupe Salazar. She received a Computer Systems Engineering degree from Instituto Tecnológico de Querétaro in 2004. She earned her M.S.E. from Universidad Autónoma de Nuevo León in 2005. Her graduate research has focused on forecasting of time series through artificial neural networks and optimization of territory design problems with multiple objectives. Her research work won relevant awards such as: *The 2005 UANL Master Thesis Award*, a university-wide award granted by UANL to the best master thesis in the Engineering, Technology, and Architecture area; *Inventive and Innovating Mexican Women: Julieta Fierro 2007 Award*, 2nd place nationwide award granted by CONACYT, Instituto Nacional de las Mujeres, AMC, IMPI, and IPN for the category of invention, academic, and research; and *Innovation and Technological Creativity 2007*, recognition given by FIME-UANL, within the LX foundation anniversary.

ACKNOWLEDGEMENTS

I am grateful to my dissertation supervisors, Prof. Roger Z. Ríos Mercado and Prof. José Luis González Velarde, for their insight, guidance and encouragement to make this work possible, and for the friendship we made during this time. Thanks to the members of my dissertation committee, Mauricio Cabrera Ríos, Rafael Caballero Fernández, and Yasmín A. Ríos Solis, for their contributions, comments, and advice, that greatly improved the quality of this research. My sincere gratitude to Julián Molina for providing me guidance in the multiobjective optimization field. I am also very grateful to all those who have, in one way or another, contributed to this research. I thank each of the faculty members and fellow graduate students of the Graduate Program in Systems Engineering at Universidad Autónoma de Nuevo León, for all those unforgettable moments we have had. Special thanks to my dearest friends in Monterrey, who made my life and my stay in this city such an exceptional and unforgettable experience. I am deeply grateful to my precious treasure, my family, with those who I shared the most special and exciting moments. Thanks to God for giving me the willingness, the skills, and the opportunity to achieve my goals.

Finally, my research work would have not been possible without a doctoral fellowship from Universidad Autónoma de Nuevo León (UANL), grant NL-2006-C09-32652, and a scholarship from the Facultad de Ingeniería Mecánica y Eléctrica (FIME) at UANL. Additionally, my participation in some national and international conferences was supported by FIME-UANL, the Mexican National Council for Science and Technology under grants SEP-CONACYT 48499-Y and SEP-CONACYT 61903, Universidad Autónoma de Nuevo León under its Scientific and Technological Research Support, grants UANL-PAICYT (CA1478-07 and CE012-09), and Tecnológico de Monterrey under research grant CAT128.

DEDICATORY

To my parents Consuelo and J. Guadalupe

to my lovely brother Javitos

to all my relatives

and to all those friends who have been with me during the storm and calm.

Abstract

In general, distribution firms have complex product distribution networks which are formed by thousands of sales points. In this kind of industry there are many interesting problems from the logistic point of view that can appear in different stages of the decision process. For instance, when a firm is starting, a first problem could be the facility location: where to install the warehouses and/or distribution centers. After that, in order to provide efficient service and to reduce total costs (i.e., production, stock and distribution costs) some questions such as how many products need to be produced, and how to deliver the products to the final customer, need to be answered.

The problem addressed in this work is motivated by a real-world application from a beverage distribution firm in the city of Monterrey, Mexico. The problem consists of finding a partition of the entire set of geographical basic units (BUs) into p territories, such that a measure of territory compactness and the maximum deviation with respect to the target number of customers are both minimized. In addition, it is required to find territories that are connected and balanced (similar in size) with respect to sales volume. A territory is connected if the set of BUs belonging to it induces a connected subgraph. This problem can be found in every distribution firm and it appears before the routing plan takes place. Having shorter routes in product distribution is a direct consequence of having compact territories in the design stage. Additionally, it is well established by the firm that compact territories reduce the number of unsatisfied customers caused by different deals offered to their customers.

This dissertation includes the study and development of new optimization models and procedures for a commercial territory design problem. The core of this work focuses on the bi-objective version of this commercial territory design problem, which has not been studied before to the best of my knowledge. For this case, an exact solution procedure and heuristic methods (GRASP and Scatter Search) are developed in this dissertation.

In addition, another area of opportunity was detected: the lack of an exact method for solving the single-objective version of this problem. So, this work contributes with an exact procedure for solving this version of the problem as well.

The proposed solution procedures were tested over a set of instances randomly generated according to the real-world cases faced by the firm. The proposed solution procedures were compared to two of the most popular and successful methods in multiobjective optimization (NSGAII and SSPMO). It was observed that over all instances tested the proposed solution procedures have better performance than the NSGA-II and SSPMO methods.

Additionally, one of the proposed heuristic was applied to a large scale real-world instance provided by the firm with excellent results. The solution found by the heuristic was significantly better than the one obtained by the company.

As a conclusion, the methods developed in this work are a significant advance to the state of the art from both the scientific and practical perspective.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

This chapter is organized as follows: Section 1.1 establishes the motivation for pursuing this research work and Section 1.2 describes the problem addressed in this dissertation. The relevance of this work is given in Section 1.3 and the research objectives are presented in Section 1.4. Finally, the organization of this dissertation is presented in Section 1.5.

1.1 MOTIVATION

Territorial design is a hard task and is very common in every enterprise dedicated to product sales and product distribution, specifically when the firm needs to divide the market into smaller regions to delegate responsibilities and to facilitate the sales and distribution of goods. These decisions need to be constantly evaluated due to the frequent market changes such as introduction of new products or changes in the workload, which are factors that affect the territory design. Additionally, the large amount of customers that need to be grouped makes this difficult task more critical. An efficient tool with capacity to provide good solutions to large problems is needed. In this sense decision sciences play an important role in the development of efficient optimization procedures to give support to the decision maker and to make this hard task easier.

Specifically, the problem addressed in this work is motivated by a real-world application from a beverage distribution firm in the city of Monterrey, Mexico. This problem belongs to the family of territory design or districting.

Single-objective versions of this problem have been studied by Ríos-Mercado and Fernández [63] and Segura-Ramiro et al. [67] from the heuristic perspective. To the best of my knowledge, the multiobjective version of this problem has never been studied in the literature, and additionally, there is not an exact solution procedure for solving the single-objective version either.

Although several approaches have appeared in the territory design literature, a few address the study of multiobjective cases which are very common in the real world. Moreover, most of these few multiobjective cases are addressed from a single objective point of view. That is, the multiple objective functions are put together into a single weighted sum function which is optimized.

Evolutionary procedures such as SPEA, NSGA-II, MOMGA, MOMGAII, and PESA, are the most popular techniques adopted by researchers in the multiobjective optimization field. However, the development of efficient constraints-handling strategies on evolutionary algorithms (EA) has proven a very challenging task. In contrast, heuristics such as Greedy Randomize Adaptive Search Procedure (GRASP) and Scatter Search (SS) widely used in constrained optimization, give us the flexibility of exploiting the problem features in order to converge to better results than those obtained by EAs when many and difficult constraints are present. Therefore, the development of GRASP and SS strategies for this multiobjective optimization problem is an area of opportunity in this research. These strategies are compared with both NSGA-II, a state-of-the-art evolutionary algorithm, and SSPMO, a state-of-the-art metaheuristic based on the SS scheme.

1.2 Problem Statement

Territory design or districting consists of dividing a set of basic units (typically city blocks, zip-codes or individual customers) into subsets or groups according to specific planning criteria. These groups are known as territories or districts. Diverse applications from different areas require a territory design. For instance, school districts, political districting, and sales territory design. Kalcsics, Nickel, and Schröder [43] present a survey of these applications. The problem addressed in this work has features that make it very unique and not addressed before to the best of my knowledge. The single-objective version of this problem was introduced by Ríos-Mercado and Fernández [63]. Different versions of this problem have been studied by Segura-Ramiro et al. [67] and Caballero-Hernández et al. [10].

Specifically, the firm wants to partition the *basic units* (blocks) of the city into a specific number of *disjoint territories* that are suitable according to their logistic, marketing, and planning requirements. The company wishes to create a specific number of territories (p) that are *balanced* with respect to each of two attributes (number of customers and sales volume). Additionally, each territory needs to be *connected*, so basic units (BUs) in the same territory can reach each other without leaving the territory. Territory *compactness* is required to guarantee that customers within a territory are relatively close to each other.

The problem is modeled by a graph $G = (V, E)$, where V is the set of nodes (city blocks) and E is the set of edges that represents adjacency between blocks. That is, a block or BU j is associated with a node, and an edge connecting nodes i and j exists if BUs i and j are located in adjacent blocks. Multiple attributes like geographical coordinates (c_x, c_y), number of customers and sales volume are associated to each node $j \in V$. It is required that each node be assigned to only one territory (exclusive assignment). In particular, the firm seeks perfect balance among territories. This means each territory needs to have the same number of customers and sales volume associated to it. Let $A = \{1, 2\}$ be the set of node activities, where 1 refers to the number of customers and 2 refers to sales volume. Let us define the size of territory V_k with respect to activity a as $w^{(a)}(V_k) = \sum_{i \in V_k}(w_i^{(a)})$, $a \in A$, where $w_i^{(a)}$ is the value associated to activity a in node $i \in V$. Hence, the target value is given by $\mu^{(a)} = \sum_{j \in V} w_j^{(a)}/p$. Another important constraint is that of connectivity, i.e., for each pair of nodes i, j that belong to the same territory, there must exist a path between them such that it is totally contained in the territory. In addition, in each territory the BUs must be relatively close to each other (compactness).

All parameters are assumed to be known with certainty. The problem consists of finding a p-partition of V according to the specific planning requirements of balancing, connectivity and compactness. For instance, the right graph in Figure 1.1 shows an 8-partition from the original instance drawn in the left graph. This partition is a feasible solution only if each $V_k \subset V$ with $k \in \{1, 2, \ldots p\}$ is compact, connected, and balanced. Due to the discrete nature of the data, a perfect balance is practically impossible. So, a tolerance parameter ($\tau^{(a)}$) that represent a relative deviation from the target value is

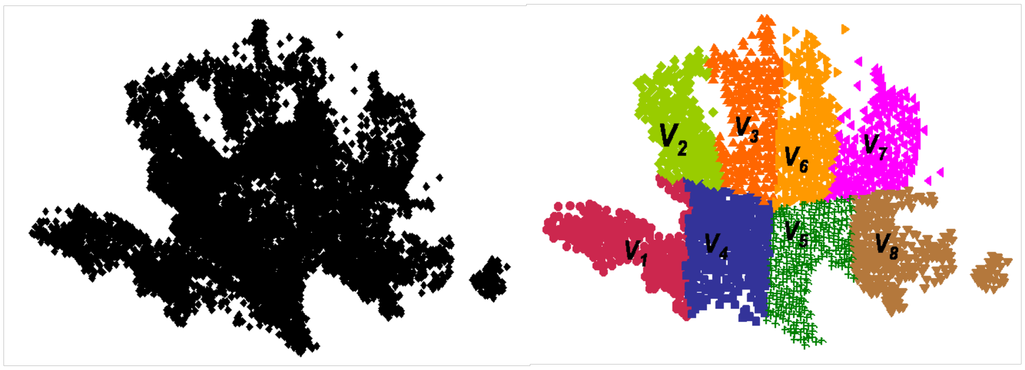

Figure 1.1: Creation of a 8-partition of V.

allowed.

For instance, suppose that the firm wishes to find a 2-partition of 14 BUs where there is a total of 42 customers, 128 units of demand, and a relative deviation $\tau(a) = 0.05$. The target value in the first activity (average number of customers) is given by $\mu^{(1)} = 21$ and the average sales volume is given by $\mu^{(2)} = 64$. Figure 1.2 presents two partition alternatives, where nodes with the same color belong to the same territory. Observe that in graph a), node l is disconnected from its territory. In contrast, in graph b) it is possible to find a path between any pair of BUs belongs to the same territory without leaving it, i.e., the connectivity constraint is satisfied. According to the balancing requirements, the territories size in design a) present a relative deviation greater than 5% from the target values. Therefore, the territory design shown in a) is not a feasible solution for the problem.

The compactness criterion is very important to the firm, because it promotes the creation of better routes during the posterior routing process. Additionally, compact territories helps to reduce the number of dissatisfied customers due to the special offers given by the firm. One can see that the number of neighboring nodes is smaller in a more compact design. Different versions of this problem are studied in this work.

1.3 Relevance

A single-objective version of this problem was introduced by Ríos-Mercado and Fernández [63]. Segura-Ramiro et al. [67] and Caballero-Hernández et al. [10] studied another vari-

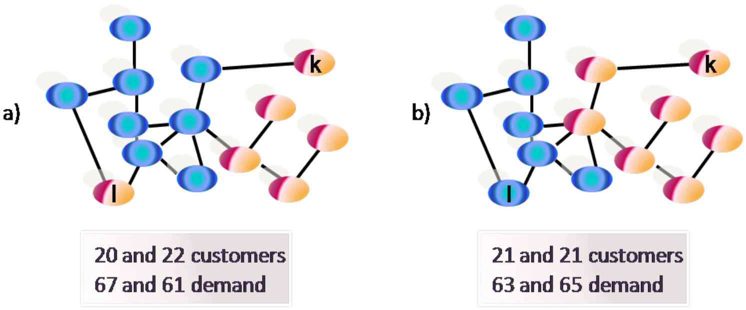

Figure 1.2: Illustration of connectivity.

ant to the single-objective problem. The basic components of this research are the study and the development of solution techniques for a class of territory design problems with a high degree of complexity.

This work introduces an exact solution procedure for the commercial territory design problem with a single-objective. To the best of my knowledge, this is the first time in which exact solutions are obtained for the single-objective version. Moreover, the core of this research focuses on the bi-objective version of this problem. That is, the commercial territory design problem is modeled as a bi-objective territory design problem in which balancing with respect to number of customers and compactness are considered as objective functions. Balancing with respect to product demand and connectivity are treated as constraints. In the bi-objective case, this research comprises both the development of exact optimization methods for solving medium-size instances, and the development of intelligent heuristics based on Greedy Randomized Adaptive Search Procedure (GRASP) and Scatter Search (SS) schemes for larger instances. The latter are developed by adequately exploiting the problem structure for providing diverse approximate *Pareto frontiers* of high quality to large instances.

The obtained solutions have the objective of giving support to the decision maker upon designing the distribution routes and the workload distribution. In addition, the partitioning permits a more efficient management of marketing offers as it reduces the number of dissatisfied customers by applying special offers in each territory. That is, efficient solutions contribute to create better route design during the routing process due to compactness (minimum dispersion) property in the territories. In addition, it provides support to the decision maker for elaborating the marketing plan and for making the best workload and resource distribution. The latter is possible because the territories are balanced with respect to both number of customers and sales volume.

1.4 OBJECTIVES

1. Development of a multiobjective model for this problem based on collaboration with industry.

2. Verification that intended optimizing criteria are indeed in conflict.

3. Development of proof of NP-completeness of this problem.

4. Design and development of exact method for finding optimal Pareto fronts to small and medium-size instances.

5. Design and development of a metaheuristic based on Scatter Search that includes the implementation of key components such as a method for generating diverse solutions, a method for solution improvement, and a method for combining solutions, which are crucial and require intelligent exploitation of problem structure.

6. Experimental verification and testing of proposed procedures.

This work contributes with the introduction of new models and procedures for solving the commercial territory design problem in its single-objective and multiobjective versions. These procedures include two variants: exact methods and heuristic procedures. For the single-objective version, the exact method is based on an iterative procedure that uses branch and bound and a cut generation scheme. The heuristic approach for this version is based on the optimization of a quadratic model through a successive dichotomies procedure. In contrast, for the multiobjective version, the exact methods are based on two variants of the ε-constraint method. The heuristic procedures includes the development of novel GRASP strategies for generating diverse solutions, which are incorporated in a Scatter Search scheme. Each component of these metaheuristic procedures is intelligently designed by taking advantage of the problem structure. An additional contribution of this work is that the proposed solution procedures are successfully applied on a large real-world instance from a beverage distribution firm.

1.5 ORGANIZATION

This dissertation is organized as follows: Chapter 2 presents an extensive literature review of diverse applications of Territory Design Problems (TDPs). Applications developed in fields such as political districting, sales territory design, services territory design, and commercial territory design are discussed in this chapter. An interesting observation is the relatively low number of applications addressed from a multiobjective perspective. Even though multiple objectives are present in the original problem, most of the time it is transformed into a single-objective problem. Commonly, the single-objective function is a weighted sum of the multiple original objectives and the solution technique is a heuristic method.

Chapter 3 introduces a new Integer Quadratic Programming (IQP) model and an exact optimization framework based on both Mixed Integer Linear Programming (MILP) and IQP models. In addition, a successive dichotomies heuristic procedure (called IQPHTDP) is proposed. It allows to obtain locally optimal solutions for large instances of the single-objective problem. The chapter includes a full evaluation of the models and the procedures.

In Chapter 4, the multiobjective version of the commercial territory design problem is described. The problem is modeled as a bi-objective MILP problem. A dispersion measure and the maximum deviation with respect to the number of customers are the objective functions that are minimized. The minimization process is subject to multiple constraints such as exclusive assignment, balancing with respect to the sales volume, and connectivity. The NP-completeness proof of the addressed problem is developed in this chapter.

Previous work in commercial territory design have been practically focused on heuristic methods. In this dissertation, an important contribution is an exact solution procedure for obtaining efficient solutions for the bi-objective version described in the Chapter 4.

Chapter 5 describes the proposed procedure. It is a combination of a cut generation procedure and the ε-constraint method. Two variants of the ε-constraint method are implemented: i) the traditional version that guarantees obtaining weakly efficient solutions, and ii) a modified improved version that assures obtaining efficient solutions. Experimental work over a set of instances shows the effectiveness of the proposed approaches.

The development of heuristic procedures for NP-hard problems is a common practice in the literature. However, even though the optimization problem addressed in this work is NP-hard, it is possible to solve medium-size instances by applying the exact procedure discussed in Chapter 5. For large instances of the problem, the exact solution procedure can not be used due to its inherent computational complexity. One can see the dramatic increase in time when attempting to solve larger instances. Therefore, the introduction of heuristic procedures is required. Chapter 6 describes the proposed multiobjective heuristic procedures. The heuristic procedures developed in this research are based on two well-known metaheuristics, GRASP and Scatter Search. Four strategies (called BGRASP-I, BGRASP-II, TGRASP-I, and TGRASP-II) based on GRASP are developed. The method based on scatter search is called SSMTDP, and it uses BGRASP-I as a diversification method. The combination method allows to obtain good and diverse solutions. That is, when combining a pair of solutions, the best features of these solutions are taken to generate three new solutions that provide diversity during the searching process.

The performance of the proposed heuristic procedures is evaluated by using different instance sets. Chapter 7 contains a detailed computational evaluation of these methods. Different metrics such as number of points, k-distance, size of space covered (SSC), and the coverage of two sets measure C(A,B) are used to evaluate the proposed procedures. These are defined in Appendix A. The evaluation includes a comparison with two of the most used multiobjective heuristics such as NSGA-II and SSPMO. NSGA-II is one of the most popular and efficient evolutionary algorithms. In contrast, SSPMO is a scatter search method that has showed successful performance over a variety of multiobjective problems. The computational work reveals the proposed GRASP procedures outperform the NSGA-II procedure, and the proposed SSMTDP outperforms SSPMO.

Conclusions, contributions, and directions for future research are highlighted in

Chapter 8. Additionally, Appendix A contains mathematical notation, basic concepts, brief description of multiobjective methods, and definitions of performance measures used in multiobjective optimization.

CHAPTER 2

RELATED WORK

Territory design or districting consists of dividing a set of *basic units* (typically city blocks, zip-codes or individual customers) into subsets or groups according to specific planning criteria. These groups are known as *territories* or districts. An extensive number of works have investigated, modeled, and developed algorithms for several applications of territory design problem (TDP). Most applications in the real world seek to satisfy more than one objective simultaneously. The problem addressed in this work is not an exception. So, a multiobjective optimization method is needed for solving the commercial territory design problem studied in this work.

Most of the TDP applications can be found in sales territories and political districting. For instance, in political districting works developed by Hess et al. [37], Fleischmann and Paraschis [29], Hojati [38], Garfinkel and Nemhauser [31], Mehrotra, Johnson, and Nemhauser[52], Bozkaya, Erkut, and Laporte [8], Ricca and Simeone [62] can be found. In sales territory design, works like the developed by Hess and Samuels [36], Marlin [50] as well as Drexl and Haase [20] can be found. For a more extensive review related to the sales territory design problem see Zoltners and Sinha [81] and for a complete survey of different applications of territory design see Kalcsics, Nickel, and Schröder [43].

This chapter includes an extensive review of the relevant work in territory design. Section 2.1 discusses single-objective models and Section 2.2 surveys multiobjective models. A summary of the most relevant work in territory design is presented in Section 2.3.

2.1 Single-Objective Territory Design Applications

Most of the works in the territory design field address single-objective models and a few works address multiobjective models. Literature review shows this fact. The analyzed works are classified according to the application addressed on each of them.

Political Districting

Hess et al. [37] present a location-allocation technique for political districting. They consider population equality, compactness, and contiguity as planning criteria. They seek to minimize the sum of squared distances between each unit and its district center. In their empirical work, their heuristic was applied to instances of up to 35 territories and 299 nodes.

Garfinkel and Nemhauser [31] present an enumeration algorithm for obtaining optimal solutions to a political districting under contiguity, compactness, and limited population deviation requirements. In their empirical work, they solved problems with 40 or fewer units.

Hojati [38] addresses a political districting problem in the city of Saskatoon, Canada. He uses a three-stage approach where Lagrangian relaxation is used first to determine district centers, then an assignment problem is solved for allocating population units to districts, and finally a sequence of capacitated transportation problems are solved for obtaining fewer splits than the existing districting plan implemented by the city.

Mehrotra, Johnson, and Nemhauser [52] address the problem of political redistricting from a column generation perspective, and present a heuristic based on branch and price for a case study in the state of South Carolina, USA. They used their method to attempt to solve a 46-county problem with 6 districts. Their method was able to obtain feasible solutions for a tolerance of 2% with respect to the target value (balancing constraint) outperforming a clustering heuristic.

Bozkaya, Erkut, and Laporte [8] address a political districting problem subject to constraints such as contiguity, population equality, compactness, and socio-economic homogeneity. They developed a tabu search procedure and used it to find solutions for a real-world case in Edmonton, Canada, with 828 basic units and 19 districts. Their procedure integrates several of the criteria into a single-objective function. Their results indicate that the algorithm produced better maps than the existing one, while improving some of the other constraints at the same time.

Bação, Lobo, and Painho [2] propose a genetic algorithm for a political districting problem. Compactness and population equality are used as objective functions. They solved the problem by optimizing the objective functions independently. In both cases connectivity is treated as a constraint. The algorithm was applied to a case provided by the Portuguese government.

Sales Districting

Hess and Samuels [36] address a sales TDP with workload balancing constraints and compactness minimization criteria. As a dispersion measure they use squared Euclidean distances. They present a heuristic based on a location-allocation scheme where a linear transportation problem is used to solve the assignment phase. Then, splits are resolved by means of a tie-breaking heuristic which assigns an area to the territory with the maximum of the area's activity. Connectivity was not considered.

Zoltners and Sinha [81] present the first review of sales territory design models. They develop a framework for sales territory alignment and several properties, which are incorporated into a general sales territory model.

Fleischmann and Paraschis [29] address a sales territory design problem arising in a German company for consumer goods. They formulate the problem as a MILP and develop a procedure based on a location-allocation approach. Specifically they have to allocate 168 sales agents in 1400 postal areas. They consider balancing workload (25% tolerance) and compactness as planning criteria. To ensure compact districts, they use weighted squared Euclidean distances as a dispersion measure. Connectivity was not considered.

Drexl and Haase [20] study the solution of sales force deployment which involves the solution of four interrelated subproblems namely: sales force sizing, salesman location, sales territory alignment, and sales resource allocation. They present a MILP model for the problem of maximizing revenue subject to connectivity and profit-related constraints for their subproblems. They propose an approximation method based on successively solving a series of MILPs.

Public Service Districting

D'Amico et al. [16] present a simulated annealing algorithm to the problem of redistricting or redrawing police command boundaries. They model the problem as a constrained graph-partitioning problem involving partitioning of a police jurisdiction into command districts subject to constraints of contiguity, compactness, convexity, and size. Since the districting plan affects urban emergency services, they also consider quality-of-service constraints. They tested their method in a case study in the Buffalo Police Department, New York. They were able to significantly reduce the officer workload disparity while maintaining current levels of response time in a 409-node network with 5 districts.

Blais, Lapierre, and Laporte [5] describe a districting study undertaken for a local community health clinic in Montreal. In their problem, a territory had to be partitioned into six districts and five disticting criteria had to be met: indivisibility of basic units, respect for borough boundaries, connectivity, visiting personnel mobility, and workload equilibrium. The last two criteria are combined into a single-objective function and the problem was solved by a tabu search technique that iteratively moves a basic unit to an adjacent district or swaps two basic units between adjacent districts. Their proposed solution procedure generates solutions at least as good as those produced by a team of experts.

Commercial Territory Design

Commercial territory design is a recent territory design application. It was introduced by Ríos-Mercado and Fernández [63]. They consider the objective function of the well-known p-Center Problem (pCP) to create compact territories. They use three different balance requirements: number of customers, sales volume, and workload. Due to the complexity of the problem, they developed a reactive GRASP procedure to solve it. Their proposed procedure outperformed the company method in both, solution quality and degree of infeasibility with respect to the balancing requirements. Different versions of this problem have been studied by Segura-Ramiro et al. [67] and Caballero-Hernández et al. [10]. In [67] they use another dispersion measure that is very common in facility location. It is the objective function of the p-Median Problem (pMP). Balancing requirements are considered as constraints. They solved the problem by an implementation of a well-known heuristic technique called location-allocation. The results showed good heuristic performance.

All previous work in commercial territory design address single-objective versions and all from the heuristic perspective. This motivates one of the contributions in this dissertation, the proposal of an exact optimization method for solving the single-objective version of this problem.

2.2 Multiobjective Territory Design Applications

Political Districting

Tavares et al. [71] study a multiobjective public service districting problem. They considered multiple criteria such as location of the zone with respect to the network, mobility structure within a zone, zone corresponding to administrative structures, centers of attraction in the zone, social nature and geographical nature. They proposed an evolutionary algorithm with local search and applied it to a real-world case of the Paris region public transportation. They discussed results for bi-objective cases considering different criteria combination.

Guo, Trinidad, and Smith [34] propose a multi-objective zoning and aggregation tool (MOZART). MOZART is an integration of a graph partitioning engine with a Geographic Information System (GIS) through a graphical user interface. They illustrated the performance of MOZART by solving two zoning problems from three government local areas in Victoria: Kingston, Bayside, and Glen Eira. The first part of their experimental work was carried out taking into account one single objective of equality in population size. In contrast, in the second part of their experimental work both equity in population and compactness were treated as objective functions. They report a case with 577 census collection districts and 20 zones, the inclusion of compactness as the second zoning objective yields zones of better shapes.

Bong and Wang [6] present a multi-objective hybrid metaheuristic approach for GIS-based spatial zoning model. Their heuristic procedure is a combination of tabu search and scatter search. They show the procedure performance by solving a political districting problem with 55 basic units and 3 districts. Equity in population, compactness, and socio-economic homogeneity are treated as objectives.

Ricca and Simeone [62] address a multiple criteria political districting problem. Such criteria were connectivity, population equality, compactness, and conformity to administrative boundaries. They transformed the multiobjective model into a single-objective model, where the objective function is a convex combination of three objective functions (inequality, noncompactness, and nonconformity to administrative boundaries), and connectivity is considered as a constraint. They compared the behavior of four local search metaheuristics (descent, tabu search, simulated annealing, and old bachelor acceptance) over a sample of five Italian regions. The old bachelor acceptance produced the best results in most of the cases.

School Districting

Bowerman, Hall, and Calamai [7] present a multiobjective approach for solving a school bus routing problem. Their proposed a heuristic technique that at first it groups students into clustering using a multiobjective districting algorithm. After that, a school bus route and the bus stops for each cluster are generated by using a combination of a set covering procedure and a traveling salesman problem procedure. They report experimental results for a real-world instance in Wellington County, Ontario. The districting algorithm considers four objectives: minimizing the number of routes, minimizing the length of the routes, load balancing, and compactness of the routes. The three last criteria are placed in a weighted objective function and the number of routes is the dominant objective, i.e., a solution with fewer routes is always favored over a solution with more. Different plans were designed using different set of weights over the optimization criteria.

Scott, Cromley, and Cromley [65] make a multiobjective analysis of school districting in a case study from Connecticut. They propose a mixed-integer goal programming model where the goal constraints are to minimize disparities in: minority enrollments, grand-list/student ratios, student-teacher ratios, and overall enrollment. The number of districts is not fixed and the contiguity criterion is not formulated in an explicit way. Experimental work using different weighting scenarios reveals that the traditional distance-minimizing or transportation-minimizing objectives are in conflict with all other aims of equity and quality of educational opportunities.

MISCELLANEOUS

Ricca [61] address a territory aggregation problem in Rome. A heuristic procedure based on an old bachelor acceptance is implemented. Compactness, population equality, and inner variance are the optimization criteria. Inner variance is used to guarantee homogeneous zones according to some socio-economic factors such as the population, number of schools, hospitals, and shopping centers. The objective function used in this work is a convex combination of the optimization criteria. Different sets of weights were used to obtain approximate efficient solutions. The heuristic technique reported better designs than the existing design.

2.3 SUMMARY

Tables 2.1 and 2.1 contain a summary of the most important work on territory design that have been developed in diverse fields such as political districting, sales districting, and public services. This table illustrates the main features included on theses applications. Planning criteria (third column) as balancing, connectivity, and fixed number of territories are shown as 'B', 'C', and 'F', respectively. In those works where the number of territory is not fixed, the capital letter 'F' is replaced by 'V', and '-' appears in the cases where connectivity is not a constraint. In the fourth column, 'Single($\sum$)' means that two or more criteria were placed together in a weighted sum objective function.

Observe that the literature reviewed on territory design reveals the following facts. Very few works address multiobjective models and all of these are basically heuristic techniques for obtaining approximate Pareto fronts. To the best of my knowledge, this work is the first to provide a method for obtaining efficient frontiers for the bi-objective territory design problem with compactness, multiple balancing, and connectivity as planning criteria. In particular, the problem studied in this work can be seen as the bi-objective extension to the model developed in [67].

Table 2.1: A) Summary of territory design applications.

Author	**Application**	**Criteria**	**Objective**	**Solution Technique**
Hess and Weaver [37]	Political	B,C,F	Single	Location-allocation
Garfinkel and Nemhauser [31]	Political	B,C,F	Single	Exact procedure
Hess and Samuels [36]	Sales	B,-,F	Single	Location-allocation
Bertolazzi et al. [4]	Services	B,-,F	Single	Exact procedure
Marlin [50]	Services	B,-,F	Single	Location-allocation
Pezzella et al. [58]	Services	B,C,F	Single	Location-allocation
Fleischman and Paraschis [29]	Sales	B,-,F	Single	Location-allocation
Hojati [38]	Political	B,C,F	Single	Location-allocation
Mehrotra [52]	Political	B,C,V	Single	Heuristic based on Branch & Price
Drexl and Haase [20]	Sales	B,C,V	Single	Heuristic
Guo et al. [34]	Political	B,C,F	Bi-objective	MOZART
Muyldermans et al. [55]	Services	B,C,F	Single($\sum$)	Heuristic of two phases
Blais et al. [5]	Services	B,C,F	Single($\sum$)	Tabu search

Table 2.2: B) Summary of territory design applications.

Author	**Application**	**Criteria**	**Objective**	**Solution Technique**
Bozkaya et al. [8]	Political	B,C,F	Single($\sum$)	Tabu search and adaptive memory
Ricca and Simeone [61]	Political	B,C,F	Single($\sum$)	Old bachelor acceptance
Bong and Wang [6]	Political	B,C,F	Three-objective	Tabu search and scatter search
Bação et al. [2]	Political	B,C,F	Single	Genetic algorithms
Chou et al. [13]	Political	B,C,F	Single($\sum$)	Simulated annealing and genetic algorithms
Tavares and Figueira [71]	Services	B,C,F	Bi-objective	Evolutionary algorithm with local search
Caballero-Hernández et al. [10]	Commercial	B,C,F	Single	GRASP
Segura-Ramiro et al. [67]	Commercial	B,C,F	Single	Location-allocation
Ricca and Simeone [62]	Political	B,C,F	Single($\sum$)	Descent, tabu search, old bachelor acceptance, and simulated annealing
Ríos-Mercado and Fernández [63]	Commercial	B,C,F	Single	Reactive GRASP

CHAPTER 3

THE SINGLE-OBJECTIVE COMMERCIAL TERRITORY DESIGN PROBLEM

Segura-Ramiro et al. [67] introduced the first single-objective mixed integer linear model (MILP) for the commercial territory design problem addressed in this research. Specifically, they considered compactness as objective function and the rest of the planning criteria were treated as constraints in their optimization model. They proposed the minimization of a dispersion measure based on the objective function of the well-known p-Median Problem (pMP). They developed a location-allocation local search heuristic that successfully handles the connectivity and the balancing constraints. The NP-completeness of this problem was established by Segura-Ramiro et al. [67]. To the best of my knowledge, no exact scheme has been developed for this problem. So, an important contribution of this research is an Iterative Cut Generation Procedure for solving TDPs (ICGP-TDP) that allows to find optimal solutions for the problem addressed in [67].

ICGP-TDP consists of iteratively solving a relaxed MILP model (relaxing the connectivity constraints), identifying violated constraints by solving an easy separation problem, and adding these violated cuts to the model. The procedure continues until optimality is reached. For the model with pMP objective, ICGP-TDP is successful in finding optimal solutions for instances with up to 150 BUs and 8 territories, and even for some cases with 200 BUs and 11 territories. In addition, a new integer quadratic programming (IQP) formulation is proposed in this work. The IQP formulation reduces the number of binary variables from n^2 to $2np$ and it allows to solve larger instances than those solved by the linear model. The ICGP-TDP framework can be applied to solve the problem by using both MILP and IQP models.

The organization of this chapter is the following. Section 3.1 contains a brief description and mixed integer linear formulations of the problem, Section 3.3 introduces the IQP formulations proposed in this dissertation. The proposed solution procedure is included in Section 3.4, and experimental work is discussed in Section 3.5. Finally, conclusions are drawn in Section 3.7.

3.1 PROBLEM STATEMENT

Given a set of city blocks or basic units (BUs), the firm wants to create a specific number of territories according to some planning criteria such as compactness, connectivity, and balancing with respect to both the number of costumers and sales volume. Due to the discrete structure of the problem and to the unique assignment constraints, it is practically impossible to have perfectly balanced territories with respect to each activity measure. Thus, in order to model balancing, a tolerance parameter $\tau^{(a)}$ for activity a is introduced. This parameter measures the relative deviation from the average territory size with respect to the activity $a \in A$. The target average is given by $\mu^{(a)} = w^{(a)}(V)/p$. Another important constraint is that of connectivity, i.e., for each i and j assigned to the same territory there must exist a path between them totally contained in the territory. In addition, in each territory the BUs must be relatively close to each other (compactness). One way to achieve this requirement is to minimize a dispersion measure. Several measures have been used in the literature. Two different measures were studied in this work, one based on the p-Center Problem (pCP) objective and the other based on the p-Median Problem (pMP) objective. This leads to two different models. Both are described below.

3.2 MIXED INTEGER LINEAR MODELS

The following notation (introduced in [67]) is used for modeling the problem.

Indices and sets

n number of blocks

p number of territories

i, j block indices; $i, j \in V = \{1, 2, \dots, n\}$

a activity index; $a \in A = \{1, 2\}$,

$a = 1(2)$ refers to the number of customers (product demand)

N^i set of adjacent nodes to node i, where

$N^i = \{j \in V, (i, j) \in E \vee (j, i) \in E\}, i \in V.$

Parameters

$w_i^{(a)}$ value of activity a in node i; $i \in V$, $a \in A$

d_{ji} Euclidean distance between j and i; $i, j \in V$

$\tau^{(a)}$ relative tolerance with respect to activity $a \in A$; $\tau^{(a)} \in [0, 1]$

Computed parameters

$\mu^{(a)} = w^{(a)}(V)/p$ average (target) value of activity a; $a \in A$

Decision variables

$$x_{ji} = \begin{cases} 1 & \text{if basic unit } j \text{ is assigned to territory with center in } i; i, j \in V \\ 0 & \text{otherwise.} \end{cases}$$

Note that $x_{ii} = 1$ implies i is a territory center.

(MPTDP) Minimize
$$z = \sum_{j \in V} \sum_{i \in V} d_{ji} x_{ji} \tag{3.1}$$

subject to
$$\sum_{i \in V} x_{ii} = p \tag{3.2}$$

$$\sum_{i \in V} x_{ji} = 1 \qquad j \in V \tag{3.3}$$

$$\sum_{j \in V} w_j^{(a)} x_{ji} \geq (1 - \tau^{(a)}) \mu^{(a)} x_{ii} \qquad i \in V; a \in A \tag{3.4}$$

$$\sum_{j \in V} w_j^{(a)} x_{ji} \leq (1 + \tau^{(a)}) \mu^{(a)} x_{ii} \qquad i \in V; a \in A \tag{3.5}$$

$$\sum_{j \in \cup_{v \in S}(N^v \setminus S)} x_{ji} -$$

$$\sum_{j\in S} x_{ji} \geq 1- \mid S \mid \qquad i \in V; \qquad S \subset [V \setminus (N^i \cup \{i\})] \quad (3.6)$$

$$x_{ji} \in \{0,1\} \qquad i,j \in V \quad (3.7)$$

Objective (3.1) represents a dispersion measure based on the pMP objective. In this sense minimizing dispersion is equivalent to maximizing compactness. Constraint (3.2) assures the creation of exactly p territories. Constraints (3.3) assure that each node is assigned to only one territory. Constraints (3.4)-(3.5) represent the territory balance with respect to each activity measure as they establish that the size of each territory must lie within a range (measured by tolerance parameter $\tau^{(a)}$) around its average size. Constraints (3.6) guarantee the connectivity of the territories. Note that there is an exponential number of such constraints.

This model was used by [67], and it can be viewed as a pMP with multiple capacity constraints, and with additional side constraints (3.4) and (3.6), respectively. Note that, when the pCP objective is used as dispersion measure the objective (3.1) is replaced by

$$z = \max_{j,i\in V} \{d_{ji}x_{ji}\}. \quad (3.8)$$

The resulting model is called CPTDP and it was introduced by Ríos-Mercado and Fernández [63].

The NP-completeness of both MPTDP and CPTDP is well established [67, 63].

Ríos-Mercado and Fernández [63] proposed a reactive GRASP to solve the CPTDP. [67] proposed a location-allocation method to solve the MPTDP. However, to the best of my knowledge no exact methods have been developed so far. Given that the connectivity constraints cannot be explicitly written out, not even commercial solvers can be applied directly. An exact solution procedure to solve MPTDP and CPTDP is proposed in this dissertation. This procedure is easily implemented under any algebraic modeler system and it can be solved by any off-the-shelf MILP solver.

Let R_MPTDP denote the relaxed model obtained by relaxing (3.6) from MPTDP. In a similar way, the relaxed model R_CPTDP is defined as the resulting model obtained by relaxing (3.6) in CPTDP.

3.3 Integer Quadratic Programming Models

The Integer Quadratic Programming (IQP) model introduced in this work reduces the number of binary variables from n^2 to $2np$. An IQP model for the pMP was proposed by [19]; however, this is the first quadratic formulation for territory design problems, to the best of my knowledge.

In this new model, let $Q = \{1, 2, \ldots, p\}$ be the set of territory indices and let y_{iq}, and z_{jq} be binary decision variables. Variables y_{iq} are used to indicate the territory centers and z_{jq} are used to represent the assigning of BUs to territories. The parameters are the same as those used in the linear model.

Decision variables for the IQP model

$$z_{jq} = \begin{cases} 1 & \text{if unit } j \text{ is assigned to territory } q;\ i \in V, q \in Q \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{iq} = \begin{cases} 1 & \text{if unit } i \text{ is the center of territory } q;\ i \in V, q \in Q \\ 0 & \text{otherwise.} \end{cases}$$

According to this definition, the equivalence between the variables in the linear model and the variables in the quadratic model is given by

$$x_{ji} = \sum_{q \in Q} z_{jq} y_{iq}$$

The resulting IQP model is the following.

$$\text{(QMPTDP)} \quad \text{Min} \quad z = \sum_{q \in Q} \sum_{j \in V} \sum_{i \in V} d_{ji} z_{jq} y_{iq} \tag{3.9}$$

$$\text{s.t.} \quad \sum_{i \in V} y_{iq} = 1 \qquad q \in Q \tag{3.10}$$

$$\sum_{q \in Q} z_{jq} = 1 \qquad j \in V \tag{3.11}$$

$$\sum_{j\in V} w_j^{(a)} z_{jq} \geq (1-\tau^{(a)})\mu^{(a)} \qquad q\in Q, a\in A \qquad (3.12)$$

$$\sum_{j\in V} w_j^{(a)} z_{jq} \leq (1+\tau^{(a)})\mu^{(a)} \qquad q\in Q, a\in A \qquad (3.13)$$

$$z_{jq} \geq y_{jq} \qquad q\in Q, j\in V \qquad (3.14)$$

$$\sum_{q\in Q} \sum_{j\in \cup_{v\in S}(N^v\setminus S)} z_{jq}y_{iq} - \sum_{q\in Q}\sum_{j\in S} z_{jq}y_{iq} \geq 1- \mid S \mid \qquad S\subset [V\setminus (N^i\cup\{i\})], \; i\in V \qquad (3.15)$$

$$z_{jq} \in \{0,1\} \qquad q\in Q, j\in V \qquad (3.16)$$

$$y_{iq} \in \{0,1\} \qquad q\in Q, i\in V \qquad (3.17)$$

The QMPTDP model uses an equivalent dispersion measure as that of MPTDP. Constraints (3.10) are to guarantee the location of only one center for each territory. Constraints (3.11) are for exclusive node assignment. The set of constraints (3.12)-(3.13) assure territory balance. Constraints (3.14) establish that BU j can not be the center of q if j is not assigned to q. The last set of quadratic constraints (3.15) guarantees connectivity. Again there is an exponential number of these constraints.

Under this quadratic formulation, a dispersion measure based on the pCP objective is given by

$$\min \quad z = \max_{i,j\in V}\left\{ d_{ji}\sum_{q\in Q} z_{jq}y_{iq}\right\}. \qquad (3.18)$$

Let QCPTDP be the resulting model when the objective function (3.9) is replaced by the dispersion measure given by (3.18).

Note that these IQP formulations are new in the literature for commercial territory design. QMPTDP is hard to solve due to the quadratic objective and quadratic connectivity constraints. Additionally, it is not possible to write these explicitly due to its exponential number. If the connectivity constraints are relaxed, the model may be solved using any MINLP method. Let R_QMPTDP be the relaxation of QMPTDP with respect to the connectivity constraints (3.15). Clearly, a solution to R_QMPTDP provides a lower bound to QMPTDP.

Note that, a feasible solution to R_QMPTDP may yield unconnected territories. One way to strengthen R_QMPTDP is to introduce the following constraints:

$$\sum_{i \in N^j} z_{iq} \geq z_{jq} \quad q \in Q; j \in V \tag{3.19}$$

These can be interpreted as follows. If j is assigned to territory q at least one of its neighbors ($i \in N^j$) must be assigned to the same territory. In this sense, these constraints avoid the unconnected subsets S with $|S| = 1$. The motivation for this stems from the fact that empirical work showed that a very large proportion of (unconnected) optimal solutions to the relaxed models R_MPTDP, R_CPTDP, R_QMPTDP, or R_QCPTDP come from subsets of cardinality equal to 1. Since there is a polynomial number of these, they can be easily incorporated into the model.

Note that, for MILP formulations the equivalent valid inequalities are given by:

$$\sum_{j^* \in N^j} x_{j^*i} \geq x_{ji} \quad i, j \in V \tag{3.20}$$

Let R1_QMPTDP be the relaxation defined by R_QMPTDP plus the additional constraints (3.19). The relaxed models for the QCPTDP model are defined in a similar way. These are called R_QCPTDP and R1_QCPTDP, respectively. Similarly, for both MPTDP and CPTDP models, new relaxed models are obtained by adding (3.20) in the relaxed models R_MPTDP and R_CPTDP, respectively. We called these R1_MPTDP and R1_CPTDP, respectively.

In the following section a solution framework is illustrated. This framework can be used to solve the problem using both MILP and IQP formulations. The proposed procedure guarantees global optimal solution for MILP models and local or global optimal solutions for IQPs, depending on what method is used for solving the relaxed subproblem.

3.4 The ICGP-TDP Procedure

One of the main difficulties for obtaining exact solutions for any of these models arise from the exponential number of connectivity constraints. The explicit enumeration of these constraints results practically impossible. Thus, to get optimal solutions an iterative procedure that uses branch and bound and a cut generation scheme is proposed. The idea is relatively simple. By relaxing the connectivity constraints, a relaxed problem that can be solved by branch and bound is obtained. The solution to this relaxed problem is checked for connectivity. Then, a separation problem is solve. This problem is polynomially solvable (see Algorithm 2). The violated valid inequalities (if any) are then added to the relaxed model and the procedure continues until no more violated inequalities are found. The ICGP-TDP procedure is outlined in Algorithm 1. For solving the MILP relaxed models, the SolveMILP method in ICGP-TDP uses any branch-and-bound method. In contrast, the SolveIQP method may call either an exact or an approximate method. In this case, in an attempt to come up with a way to find faster solutions, a local optimum method was used for finding good feasible solutions for the IQP relaxed models. An issue to investigate is precisely the trade-off between time and solution quality.

Algorithm 1 ICGP-TDP (*P, DispMeasure, ModelType*)

```
Input:
  P:= Instance of the TDP problem
  DispMeasure:= $p$CP or $p$MP objective function
  ModelType:= MILP or IQP
Output: $X = (X_1, X_2, \ldots, X_p)$:= A feasible $p$-partition of V
  Cuts ← $\emptyset$ {Cut set}
  Model← GenerateRelaxedModel(P, DispMeasure, ModelType)
  while (Cuts$\neq \emptyset$) do
    if (ModelType = MILP) then
      $X$ ← SolveMILP(Model)
    else
      $X$ ← SolveIQP(Model)
    end if
    Cuts ← SolveSeparationProblem(P,$X$)
    AddCuts(Model, Cuts)
  end while
  return $X$
```

3.4.1 THE SEPARATION PROBLEM

Suppose we have a graph $G = (V, E)$ and a p-partition $X = (X_1, X_2, \ldots, X_p)$, where each of these sets X_k induces a subgraph $G_k = (X_k, E(X_k))$ of G and a center $c_k \in X_k$. The separation problem consists of identifying all connected components of G_k. Each of the connected components of G_k that does not contains the center c_k is used to generate a violated connectivity constraint in the problem. Algorithm 2 describes the steps to solve the separation problem. Note that Step 3 can be efficiently done by breadth first search.

Algorithm 2 SolveSeparationProblem procedure.

```
Input:
  P:= Instance of the TDP problem
  $X = (X_1, X_2, \ldots, X_p)$:= A $p$-partition of V
Output: (Cuts)
  Cuts := Set of violated connectivity constraints
  Cuts ← $\emptyset$
  for ($k = 1, \ldots, p$) do
    Obtain connected components $S_1, S_2, \ldots, S_t$ of $G(X_k, E(X_k))$
    For each $S_t$ such that $c_k \notin S_t$ generate the violated cut and add it to Cuts
  end for
  return Cuts
```

To illustrate the separation problem consider an example with $n = 11$ nodes and $p = 2$ territories (depicted in Figure 3.1). The doted lines represent the nodes belonging to the same territory. Suppose that a solution to the relaxed problem after applying branch and bound is given by the 2-partition $X_1 = \{1, 4, 6, 7, 11\}$ with center in $c_1 = 4$ and $X_2 = \{2, 3, 5, 8, 9, 10\}$ with center in $c_2 = 5$, then the variables have the following values:

$x_{14} = x_{44} = x_{64} = x_{74} = x_{114} = 1,$

$x_{24} = x_{34} = x_{54} = x_{84} = x_{94} = x_{104} = 0,$

$x_{25} = x_{35} = x_{55} = x_{85} = x_{95} = x_{105} = 1,$

$x_{15} = x_{45} = x_{65} = x_{75} = x_{115} = 0;$

$x_{jk} = 0, \forall j, k \in V, k \neq \{4, 5\}.$

Given this solution, the separation problem (Algorithm 2) is solved to identify the connectivity constraints violated by this solution. According to the connectivity constraints (3.6), the connected components $S_1, \ldots, S_t$ are identified on each territory. As

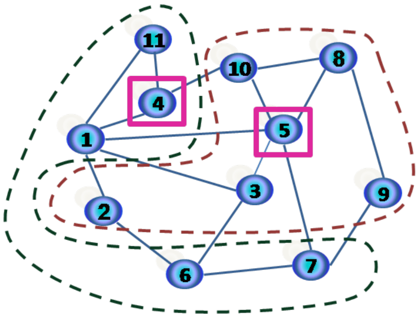

Figure 3.1: Example of an unconnected territory design for $p = 2$ and $n = 11$.

can be seen from X_1 the connected component $S_1 = \{6, 7\}$ is unconnected from c_1 then it induces a violated constraint which is generated as

$$x_{24} + x_{34} + x_{54} + x_{94} - x_{64} - x_{74} \geq -1.$$

Similarly, in X_2 the connected component $S_1 = \{2\}$ is unconnected from c_2 and the violated constraint is given by

$$x_{15} + x_{65} - x_{25} \geq 0.$$

Following the ICGP-TDP procedure, the cuts are added to the relaxed model and it is solved again. We proceed iteratively until the final solution gives us a connected territory design or a feasible solution is not found. The latter means the original problem is infeasible. Note that the process for identifying unconnected subsets on each territory can be efficiently done by *Breath First Search* (BFS).

The proposed procedure guarantees obtaining an optimal solution when the MILP problem is feasible. In addition, on each iteration, the number of aggregated cuts is equal to the total unconnected subsets identified from the given solution.

Observe that the number of (binary) decision variables in the MILP models is equal to n^2. In contrast, the IQP models have only $2np$ binary variables.

3.5 COMPUTATIONAL RESULTS

The proposed ICGP-TDP method was coded in C++ and compiled with the Sun C++ 8.0 compiler. The MILP relaxations are solved by the CPLEX 11.2 callable libraries and the IQP relaxations are solved by DICOPT [47]. Two stopping criteria were used, optimality gap ($gap \leq 5 \times 10^{-6}$) and time limit (7200 seconds). To speed up convergence the use of priorities on the binary variables was done to ensure that x_{ii} are branched before than $x_{ji}, i \neq j,\ i, j \in V$. Randomly generated instances based on real-world data provided by the industrial partner were used. Each instance topology was randomly generated as a planar graph. We considered a tolerance $\tau^{(a)} = 0.05, a \in A$, and generated three different instance sets as $(n,p) \in \{(60,4),(80,5),(100,6)\}$. For each of these sets, 20 different instances were generated. Additionally, two larger instance sets were generated for $(n,p) \in \{(150,8),(200,11)\}$, where 10 different instances were generated for each of them.

3.5.1 EVALUATION OF MILP MODELS

We first evaluate linear models CPTDP and MPTDP when the relaxed models R_CPTDP and R_MPTDP, respectively, are used within the ICGP-TDP procedure.

Tables 3.1 and 3.2 show the results for CPTDP and MPTDP, respectively. The first column indicates the instance size tested. The second column shows the percentage of instances that were solved at the first iteration (out of 20 except for the set $(150,8)$), that is, the percentage of instances that did not find any unconnected territory in the first iteration. The third column contains the average and the maximum number of iterations per instance required by the algorithm to find the optimal solution. The fourth column displays the percentage of instances solved within the specified time limit. The fifth column shows the average and the maximum number of cuts added per instance solved. Finally, the last column displays information about the CPU time (average and maximum) used per instance.

For model CPTDP, Table 3.1 indicates that a very small proportion of the instances were solved at the first iteration. As many as 26 iterations and 82 cuts were needed in the

Table 3.1: Results for CPTDP under the R_CPTDP relaxation.

size	Solved at	Iterations		Solved	Cuts/inst		Time (sec)	
(n, p)	1st iter (%)	Ave	Max	(%)	Ave	Max	Ave	Max
(60,4)	20	5.3	26	100	12.1	82	381	1446
(80,5)	10	5.4	14	90	12.4	43	2682	7200
(100,6)	10	2.3	11	40	3.5	32	5812	7200
(150,8)	0	-	-	0	-	-	7200	7200

Table 3.2: Results for MPTDP under the R_MPTDP relaxation.

size	Solved at	Iterations		Solved	Cuts/inst		Time (sec)	
(n, p)	1st iter (%)	Ave	Max	(%)	Ave	Max	Ave	Max
(60,4)	80	1.4	6	100	0.5	5	7	33
(80,5)	70	1.4	4	100	0.5	4	53	235
(100,6)	75	1.4	4	100	0.5	4	95	438
(150,8)	75	1.8	5	80	1.6	6	1900	7200

worst case. At the end of the procedure, all 100 % instances of the (60,4) were solved optimally. 90 % of the (80,5) set were solved optimally. However, the procedure struggles with the larger sets. For the two smaller sets, around 5 iterations and 12 cuts were needed on average. Note that, for a specific iteration the separation problem has the property to identify more than one unconnected subset and it generates all violated connectivity constraints at the same iteration. Note that for the (150,8) set, the procedure was unable to terminate a single iteration within the time limit.

These statistics improve significantly for the MPTDP model (Table 3.1). Except for a very few cases in the largest set, all other instances were solved optimally. A large proportion of these were solved at the very first iteration. On average it required less than 2 iterations and a very few cuts for obtaining optimal solutions. This suggests not only that the LP relaxation of the median-based model is tighter that the one of the center-based model, but solutions to the R_MPTDP relaxation yield near-connected solutions. This has a positive impact in overall solution time.

Another issue to investigate is to whether or not the introduction of constraints (3.20) have a positive effect on strengthening the model. Recall that constraints (3.20) eliminate unconnected subsets of size 1. So, in this experiment the very first relaxation was solved only for every instance and tallied the cardinality of all unconnected subsets for

Table 3.3: Size of unconnected subsets for the R_CPTDP relaxation.

size	Cuts	% cuts with			
(n,p)	identified	$\|S\| = 1$	$\|S\| = 2$	$\|S\| = 3$	$\|S\| \geq 4$
(60,4)	44	72.7	18.2	4.5	4.5
(80,5)	65	58.5	20	40	13.8
(100,6)	103	67	11.7	7.8	13.6
(150,8)	-	-	-	-	-

Table 3.4: Size of unconnected subsets for the R_MPTDP relaxation.

size	Cuts	% cuts with		
(n,p)	identified	$\|S\| = 1$	$\|S\| = 2$	$\|S\| = 3$
(60,4)	4	100	0	0
(80,5)	6	83	17	0
(100,6)	5	80	20	0
(150,8)	6	83	17	0

both CPTDP and MPTDP. A summary of this experiment is shown in Tables 3.3 and 3.4. Table 3.3 shows that most of the identified cuts for CPTDP correspond to unconnected subsets of cardinality equal to 1. For the (60,4), (80,5), and (100,6) sets, the proportion of unconnected subsets of cardinality 1 is 72.7, 58.5, and 67.0, respectively. This proportion is even more dramatic for MPTDP (see Table 3.4). One can see that the number of total unconnected subsets is considerable smaller than that of the R_CPTDP relaxation. This confirms that the MPTDP model not only has a better LP relaxation, but it also favors connectivity, which is a very important issue. Hence, these results clearly justify and motivate the introduction of the valid inequalities given by 3.20 into the relaxed models.

The following experiment clearly illustrates this issue. We now solve model MPTDP under two different relaxations: R_MPTDP and R1_MPTDP (incorporating the valid inequalities). We identify these with prefix R and R1, respectively. Table 3.5 displays the results. The second and third columns show the number of instances (out of 20) that were solved optimally at the very first iteration, that is, by solving the first relaxed models for R_MPTDP and R1_MPTDP, respectively. The fourth and fifth columns display the total number of cuts added during the execution of the algorithm. The last two columns show the percentage of instances that were optimally solved. As can be seen, relaxation R1 provides a more attractive choice in all senses. Therefore, the introduction of constraints

Table 3.5: Comparison of relaxations R_MPTDP and R1_MPTDP.

size	Solved at 1st		Cuts		solved	
(n,p)	iteration (%)		added		(%)	
	R	R1	R	R1	R	R1
(60,4)	80	100	9	0	20	100
(80,5)	70	95	9	1	20	100
(100,6)	75	90	9	2	20	100
(150,8)	5	45	6	3	80	90

(3.20) into the relaxed model provides a stronger LP representation of model MPTDP. This has indeed a positive impact in solution times.

3.5.2 Evaluation of IQP models

The IQP formulations QCPTDP and QMPTDP are considered under the R_QCPTDP and R_QMPTDP relaxations, respectively. In a similar fashion as carried out with the linear models, the distribution of the cardinality of the unconnected subsets was analyzed, when only the very first relaxation is solved. Tables 3.6 and 3.7 display the results for QCPTDP and QMPTDP, respectively. The description is similar to that of Table 3.3. It can be seen that most of the unconnected subsets have cardinality 1, which is a similar behavior observed in the linear models. Another observation is that the relaxation of the median-based model provides solutions with a higher degree of connectivity that the one provided by the center-based model. So a considerable less amount of effort will be needed to eventually solved a median-based model with connectivity constraints. These results clearly motivate the introduction of valid inequalities (3.19) into the relaxed models.

Table 3.6: Size of unconnected subsets for the R_QCPTDP relaxation.

size	Cuts	% cuts with			
(n,p)	identified	$\|S\| = 1$	$\|S\| = 2$	$\|S\| = 3$	$\|S\| \geq 4$
(60,4)	662	68	21	6	5
(80,5)	956	73	17	6	4
(100,6)	1340	77	17	4	2
(150,8)	1088	82	14	3	1

The effect of incorporating constraints (3.19) into the relaxed model R1_QMPTDP is evaluated. Table 3.8 shows the results when QMPTDP is solved under the R1_QMPTDP

Table 3.7: Size of unconnected subsets for the R_QMPTDP relaxation.

size	Cuts	% cuts with		
(n,p)	identified	$\|S\|=1$	$\|S\|=2$	$\|S\|=3$
(60,4)	3	100	0	0
(80,5)	5	40	20	40
(100,6)	6	67	33	0
(150,8)	5	60	40	0

Table 3.8: Solution of QMPTDP under the R1_QMPTDP relaxation.

size	Solved at	Cuts	Iterations		
(n,p)	1st iter (%)	added	Min	Ave	Max
(60,4)	95	1	1	1.1	2
(80,5)	85	4	1	1.2	2
(100,6)	95	1	1	1.1	2
(150,8)	100	0	1	1.0	1

relaxation. The second column shows the percentage of instances that were solved at the very first iteration. The third column display the total average number of cuts added. Columns 4 through 6 gives information on the number of iterations needed to reach optimality. As can be seen, the addition of constraints (3.19) gives excellent results as very little additional effort in both cut generation and number of iterations was needed.

When attempting to carry out a similar experiment for the QCPTDP model under the R1 relaxation, it was observed that the LP relaxation was still extremely weak. The procedure could not terminate a single iteration within the specified time limit. The effect of adding the cuts resulted in even higher running times. So, clearly this effort did not pay off.

3.5.3 Comparing MILP and IQP

Clearly, it has shown that solving the quadratic models is faster than solving the linear models. However, solving the quadratic model with local-optimum methods no longer assures global optimality. Therefore, an important issue to investigate is precisely the trade-off between solution quality and computational effort. We apply the solution procedure to models MPTDP and QMPTDP on two instance data sets (60,4) and (150,8). Results are shown in Table 3.9 and 3.10, respectively. The fourth column shows the rela-

Table 3.9: Comparison of MPTDP and QMPTDP models for instance set (60, 4).

	Objective value		Gap (%)	Time (sec)	
Inst	MPTDP	QMPTDP		MPTDP	QMPTDP
1	5305.57	5306.00	0.01	4	2
2	5451.68	5463.00	0.21	4	2
3	5507.88	5553.00	0.82	10	2
4	5935.67	6114.00	3.00	4	6
5	5303.20	5303.20	0.00	3	2
6	5253.94	5280.00	0.50	33	3
7	5460.18	5855.00	7.23	4	3
8	5309.96	5314.00	0.08	4	2
9	5224.51	5225.00	0.01	2	3
10	5350.15	6140.00	14.76	3	2
11	5150.91	5152.00	0.02	3	2
12	5597.50	5705.00	1.92	6	2
13	5731.99	5732.00	0.00	3	3
14	5462.96	5869.00	7.43	5	2
15	5332.77	5759.00	7.99	6	2
16	5399.54	5499.00	1.84	14	2
17	5602.86	5603.00	0.00	3	2
18	5773.96	6299.00	9.09	4	4
19	5543.45	5544.00	0.01	17	2
20	5767.54	5768.00	0.01	4	2

tive optimality gap of the solution found under the quadratic model (that is, with respect to the optimal solution found by the linear model). For the instances marked with a star (*), the MILP could not find an optimal solution within the specified time limit so a best integer solution is used instead.

As it can be seen from Table 3.9, 19 out of 20 instances found with the quadratic model fall within 10% of the optimal solution, and 60% of the solutions lay within 1% of optimality. Time is not an issue in these sets as can be seen in the last two columns. However, for the larger instances (displayed in Table 3.10), time becomes important. We can see how time significantly increases for the MILP model. There are two instances where time limit was reached when using the MILP model. When using the quadratic model, all instances were solved within 1 minute of CPU time, delivering optimality gaps of less than 5% in 90% of the instances. So this makes the quadratic model a very attractive choice for relatively large instances.

Table 3.10: Comparison of MPTDP and QMPTDP models for instance set (150, 8).

	Objective value		Gap (%)	Time (sec)	
Inst	MPTDP	QMPTDP		MPTDP	QMPTDP
1	9511.76	9979	4.91	1137	9
2	(*) 9404.60	9509	1.11	7200	29
3	9125.61	9130	0.05	90	32
4	9359.00	9646	3.07	147	30
5	9506.58	10494	10.39	455	42
6	9039.06	9088	0.54	78	25
7	9819.18	10017	2.02	1842	29
8	(*) 9202.13	9550	3.78	7200	34
9	9670.90	9972	3.11	730	28
10	9570.58	9794	2.33	125	26

Table 3.11: Time comparison for QMPTDP and MPTDP models.

size	MPTDP time (sec)			QMPTDP time (sec)		
(n, p)	Min	Ave	Max	Min	Ave	Max
(60,4)	2	6.8	33	2	2.5	6
(80,5)	8	53.2	235	4	5.6	12
(100,6)	18	94.8	438	7	8.8	23
(150,8)	78	1900.4	7200	9	28.4	42

A summary of computational effort and solution quality over four different data sets is shown in Tables 3.11 and 3.12, respectively. These tables show that the CPU time employed for solving the quadratic model is relatively low compared with the time used by the linear model. Furthermore, the average relative optimality gaps for the quadratic model are less than 4%. In many cases the solution to the quadratic model was less than 1%.

Moreover, it was attempted to solve instances from a largest set (200,11) by using both MPTDP and QMPTDP models. In this case, the ICGP-TDP procedure reported

Table 3.12: Solution quality for QMPTDP.

size	Gap (%)		
(n, p)	Min	Average	Max
(60,4)	0.00	2.75	14.80
(80,5)	0.01	2.61	8.15
(100,6)	0.06	3.14	7.56
(150,8)	0.05	3.13	10.39

optimal solutions for 4 out of 10 instances tested (using MPTDP model). In contrast, ICGP-TDP reported locally optimal solutions for 8 out of 10 instances tested by using the QMPTDP model. Additionally, an instance with $\tau^{(a)} = 0.05, a \in A$; $n = 280$ and $p = 9$ was generated. This instance was tested using the MPTDP formulation and in the first relaxed model (R_MPTDP), the branch and bound reported a percentage of relative optimality equal to 14.48%, after 24 hours. The same instance was tested by using R1_QMPTDP and ICGP-TDP reported a connected solution in less than 4 minutes. Comparing the objective value for QMPTDP with the best lower bound found by branch and bound, it reached a relative optimality of 12.17%. Finally, it was even tested the ICGP-TDP procedure (with QMPTDP model) for instances with $n = 500$ and $p = 20$ and it was observed that is possible to find locally optimal solutions to theses cases. Therefore, the QMPTDP model is a fast and attractive alternative to find relatively good solutions for large instances. It offers a good compromise between time and quality.

3.6 THE IQPHTDP PROCEDURE

During the experimental work for the IQMPTDP model, empirical results showed that the QMPTDP model allows to obtain locally optimal solutions for larger instances than the MILP model. In an attempt to exploit this, a new heuristic procedure, called IQPHTDP is introduced in this dissertation. The procedure consists basically of a successive dichotomies process, where a (parent) problem is divided into two (children) subproblems which are solved in an independent way. If the number of BUs in a children subproblem is larger than the *maxN* parameter, this subproblem is divided in two smaller subproblems, and so on. The process of successive dichotomies stops when the resulting subproblems are small enough to be solved (exactly or approximately) by a MINLP method. For instance, MINLP methods such as DICOPT [47] or AlphaECP [76] can be used. The final solution for the original instance is formed by those partitions obtained by solving the smaller subproblems. The IQPHTDP procedure can be easily implemented. In this work it was coded in C++ and it calls to ICGP-TDP (see Algorithm 1) procedure for solving the IQP subproblems.

Computational results showed that, IQPHTDP is an attractive technique for obtaining locally optimal solutions for instances with more than 500 nodes. However, its performance depends on a control parameter ρ which helps to obtain feasible solutions or at least to reduce the infeasibility value with respect to the balancing constraints. When IQPHTDP reports an infeasible solution, a simple local search procedure can be applied to reach feasibility. The infeasibility can be easily eliminated by applying a simple local search procedure. The procedure was successful for solving the single-objective version of the commercial territory design problem. However, it is not attractive for solving the bi-objective version of the problem.

Algorithm 3 shows schematically the proposed solution procedure. It consists basically of solving a series of IQP models in such a way that those problems with more BUs than the number allowed by the *maxN* parameter are solved using a p value equal to 2. It means, given a TDP instance, if $|V| >$ *maxN* the algorithm carries out a dichotomy of this instance by solving the problem with $p = 2$. For example, suppose the size of the original instance is given by $(n, p) =$ (1000,49), let $S = (V_1, V_2)$ be the solution of the original instance with $p = 2$. Then the target size for V_1 should be equal to the size determined by the target value of 24 territories from the original instance, and the target size for V_2 should be equal to the target value of 25 territories from the original instance. It is, this dichotomy yields two smaller subproblems and each of them is analyzed to determine if another dichotomy is required or not. If the instance (subproblem) given by V_1 is such that $|V_1| <$ *maxN*, then the subproblem is solved by using $p =$ 24, and the target value for each territory is given by the target value $\mu^{(a)}$ and the tolerance $\tau^{(a)}, a \in A$ (obtained from the original instance). In another case, the iterative process of successive dichotomies continues until all subproblems are solved with $|V| <$*maxN*. The final solution is obtained by joint all partitions obtained by solving the smaller subproblems.

Observe that, IQPHTDP requires a TDP instance, *maxN*, and ρ as input specified by the user. The control parameter ρ helps to keep balanced partitions as much as possible. It is required because if the initial dichotomy produces a partition with high relative deviation with respect to the average (target value), in the following dichotomies this value affects in such a way that the final subproblems could not have a feasible solution with respect to the average size in the original instance.

3.6.1 Example of IQPHTDP

Suppose that IQPHTDP is used for solving an instance I with (n, p) = (1999,50) and the input parameters are $maxN$ = 300, and $\rho = 0.8$. Figure 3.2 shows the dichotomies process, observe that, in the first dichotomy each partition V_1' and V_2' contains a half of the total number of required territories (it is 25 out of 50) and the number of BUs on each of them are greater than $maxN$, so another dichotomy is needed. Partitions V_1' and V_2' are used to generate two subproblems of TDP $((G_1' = (V_1', E(V_1'))) \subset G$, and $(G_2' = (V_2', E(V_2'))) \subset G$ respectively) which are solved using $p = 2$. In Figure 3.2, (V_3', V_4') corresponds to the 2-partition of V_1', and (V_5', V_6') is a 2-partition of V_2'. These partitions V_3', V_4', V_5', and V_6' contain more BUs than the allowed by $maxN$, so the dichotomic process is applied on each of them until the last obtained partitions $V_l' : l = 7, ..., 14$ contain less BUs than the limit value (given by $maxN$). The latter are solved using the number of territories contained on each partition. For instance, the subproblem given by V_7' is solved for $p_7' = 6$ and the subproblem given by V_8' is solved for p_8' = 6. The upper and lower balancing requirements are taken from the original instance I. Note that, the balancing requirements for dichotomies are computed using the control parameter ρ and the number of territories contained on each sub-instance (see Algorithm 3).

The final solution for instance I is computed by putting together all partitions obtained for solving the small subproblems (in the example the small subproblems are those generated by $V_l' : l = 7, ..., 14$). Figure 3.3 shows the final solution obtained for instance I by applying of IQPHTDP procedure.

Some small subproblems can be infeasible with respect to the balancing constraints, so the solution for the original instance will be infeasible. This can be avoided by selecting a suitable value for the ρ parameter. In another case, a simple local search procedure can be applied to the final solution given by the IQPHTDP procedure in order to reach a feasible solution.

```
**Algorithm 3** IQPHTDP(*I, maxN, ρ*)
Input:
  $I$:= Instance of TDP
  $maxN$:= Maximum number of BUs for solving the IQP model
  $\rho$:= Control parameter
Output:  $S = (V_1, ..., V_p)$: Solution, $p$-partition of $V$
  $I_0(n_0, p_0, V_0, w_i^{(a)}, \tau^{(a)}) = I$:= Original instance
  $L = \emptyset$:= Subproblems list
  $L = L \cup I_0$, $c = 0$
  while ($L$ has instances to be solved) do
    Take $I_c \in L$
    if ($n_c > maxN$) then
      The target value for those territories contained in $V_c$ is given by
          $$\mu_{V_c}^{(a)} = \frac{\sum_{1 \in V_c} w_i^{(a)}}{p_c}$$
      if ($p_c$) is pair then
        $p'_{c1} = \frac{p_c}{2}$; $p'_{c2} = \frac{p_c}{2}$
      else
        $p'_{c1} = \frac{p_c+1}{2}$; $p'_{c2} = \frac{p_c-1}{2}$
      end if
      Solve $I_c$ for $p = 2$
      Let $S_c = (V'_{c1}, V'_{c2})$ be the obtained solution from $I_c$
      the size for $V'_{c1}$ should be,
          $$p'_{c1}(1 - \rho\tau^{(a)})\mu_{V_c}^{(a)} \geq \sum_{i \in V_c} z_{i1} \leq p'_{c1}(1 + \rho\tau^{(a)})\mu_{V_c}^{(a)}$$
      and the size for $V'_{c2}$ should be,
          $$p'_{c2}(1 - \rho\tau^{(a)})\mu_{V_c}^{(a)} \geq \sum_{i \in V_c} z_{i2} \leq p'_{c2}(1 + \rho\tau^{(a)})\mu_{V_c}^{(a)}, a \in A$$
      Add the instances defined by $V'_{c1}$ and $V'_{c2}$ in $L$. It is,
      $L = L \cup Instance(V'_{c1}) \cup Instance(V'_{c2})$
    else
      Solve $I_c$ for $p = p_c$ and $\mu^{(a)}$ {It uses the target value associated to the instance I}
    end if
    $c = c + 1$
  end while
  Put together all partitions obtained from instances with $n_c < maxN$
  return $S = (V_1, ..., V_p)$
```

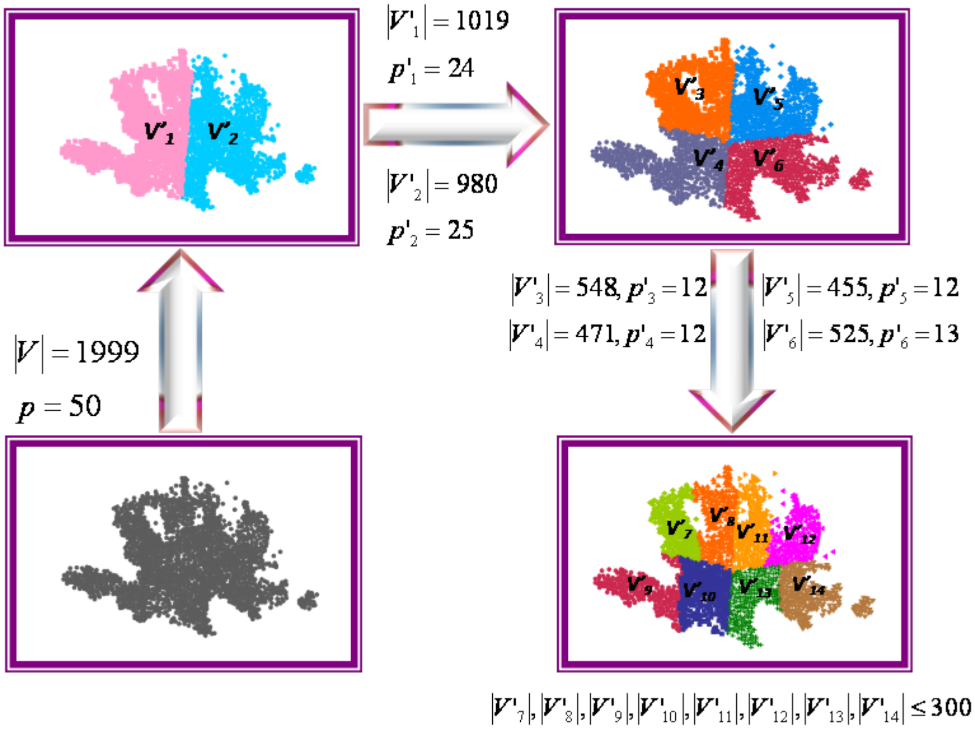
$|V'_1| = 1019$
$p'_1 = 24$
$|V'_2| = 980$
$p'_2 = 25$
V'1
V'2
V'3
V'4
V'5
V'6
$|V'_3| = 548, p'_3 = 12$
$|V'_4| = 471, p'_4 = 12$
$|V'_5| = 455, p'_5 = 12$
$|V'_6| = 525, p'_6 = 13$
$|V| = 1999$
$p = 50$
V'7
V'8
V'9
V'10
V'11
V'12
V'13
V'14
$|V'_7|, |V'_8|, |V'_9|, |V'_{10}|, |V'_{11}|, |V'_{12}|, |V'_{13}|, |V'_{14}| \leq 300$

Figure 3.2: Successive dichotomies process for solving the instance I.

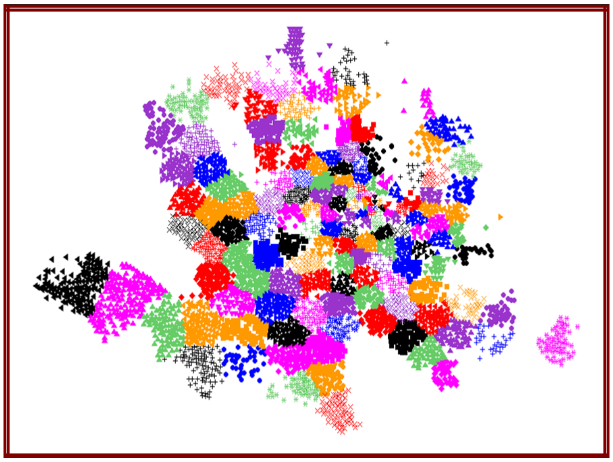

Figure 3.3: Final solution for instance I (using IQPHTDP).

Table 3.13: Summary of instances tested by different values of ρ.

(n, p)	ρ	Feasible
$(2000, 50)$	0.2	9
$(2000, 50)$	0.1	7
$(1000, 50)$	0.1	10
$(2000, 50)$	1.0	0
$(1000, 50)$	1.0	0

3.6.2 Experimental Work

The goal of this part was to evaluate the performance of the IQPHTDP procedure. The experimental work was carried out over two instance sets $(n, p) \in \{$(2000,50),(1000,50)$\}$ with $\tau^{(a)} =$ 0.05, for each of them 10 instances were randomly generated. Different values of ρ were used in order to determine the effect yielded by this parameter in the final solution reported by the IQPHTDP procedure. A summary of this experimental work is shown in Table 3.13, first column contains the instance size, second column shows the control parameter value (ρ), and third column displays the number of those instances in which the IQPHTDP procedure reported a feasible final solution. When $\rho = 1$, it means that the balancing deviation in all IQP subproblems is given by $\tau^{(a)}$. It implies that, when the size of a partition is really close to the balancing bounds, the following partitions created from this partition may be really unbalanced with respect to the target value in the original instance. In contrast, small values of ρ are very restrictive during the successive dichotomies process and this fact permits to obtain locally optimal solution for the original instance.

During the experimental work, it was observed that the computational effort of the IQPHTDP procedure is relatively large. The worst case for solving the tested instances required about 3 hours of CPU, and some times the obtained final solution was highly infeasible. When the final solution is infeasible, the IQPHTDP procedure can be applied by using another ρ value, however, this change does not guarantee that the new solution will be feasible and the time increases for each trial-and-error attempt of the ρ value. A local search procedure is another option for attempting to reach feasibility. The last option increases the time as well.

3.7 Conclusions

In this chapter, MILP and IQP models for the commercial territory design problem with connectivity and multiple balancing constraints are described. The IQP formulations use a significantly fewer number of binary variables than the MILP formulations. The IQP models are new in the literature of territory design.

An exact solution procedure (ICGP-TDP) based on branch and bound and a cut generation strategy was introduced in this chapter. This method can be applied to both MILP and IQP models.

The models were strengthen by the introduction of valid inequalities that eliminate unconnected subsets of size 1. We have observed empirically that most of the unconnected subsets found in the relaxed models (relaxing the connectivity constraints) have cardinality equal to 1, so this motivates the introduction of these valid inequalities. We empirically proved that the cut did in fact helped find connected territories faster.

When the solution method was applied to solve instances under linear and quadratic models, the proposed IQP models showed a balanced behavior between quality and effort. For the larger instances, execution times under the quadratic models were significantly lower than those observed under the linear models. The solution quality of those obtained by the quadratic model over all instances was in the range of 0.0 to 14.8%, and in most cases, less than 5%.

It was observed that the pMP objective is more LP-friendly than the pCP objective. During the branch and bound process the linear relaxation for pMP objective showed better behavior than the linear relaxation for the pCP objective. Furthermore, it was also observed that solutions obtained from the relaxation of the median-based models had a very high degree of connectivity. This had a very good impact on computational efficiency since very few iterations were needed to find connected solutions as opposed to the center-based models. So, in the absence of a standard dispersion measure, the pMP objective is a good choice to be used in other territory design problems that have compactness as performance measure.

In the experimental work, instances with up to 150 BUs and 8 territories were efficiently solved by using MILP models. To the best of my knowledge, in general territory design the largest instance with connectivity constraints solved optimally had no more than 50 BUs [31]. As far as this particular commercial TDP is concerned, the proposed method is the first exact optimization scheme developed for the problem.

For IQPs models, locally optimal solutions for instances with up to 500 BUs and 20 territories were obtained. This instance size was intractable under MILP formulations. One of the advantages of the proposed approach is that it can be implemented relatively easy with off-the-shelf MILP and MINLP solvers.

An additional heuristic procedure called IQPHTDP was proposed and described in Section 3.6. This procedure allows to obtain locally optimal solutions for large instances (1000 and 2000 BUs) of the problem addressed in this chapter. These instances were intractable by using the exact method. However, the performance of this procedure depends on the choice of the control parameter ρ. Bad values of ρ may yield highly infeasible solutions with respect to the balancing requirements.

It is clear that the IQPHTDP procedure is not an attractive choice for embedding into an exact method such as the ε-constraint method, for the bi-objective model. This stems from the fact that in the ε-constraint method,a single-objective model has to be solved many times, which would yield very high execution times.

CHAPTER 4

THE MULTIOBJECTIVE COMMERCIAL TERRITORY DESIGN PROBLEM

In the literature of territory design very few works address multiobjective territory design problems and all of these are basically heuristic techniques for obtaining approximate Pareto fronts. In particular, a bi-objective model for a commercial territory design problem is introduced in this dissertation. This can be seen as the bi-objective extension to the model developed in [67]. In addition, the completeness proof for this problem is developed in this work.

This chapter is organized as follows. Section 4.1 presents a problem description, mathematical formulation is included in Section 4.2, and Section 4.3 contains the complexity proof for the problem addressed in this research.

4.1 PROBLEM STATEMENT

Given a set V of city blocks or *basic units* (BUs), the firm wishes to partition this set into a fixed number (p) of disjoint territories that are suitable according to some planning criteria. The territories need to be balanced with respect to each of two different activity measures (number of customers and sales volume). Additionally, each territory has to be connected, so the set of BUs belonging to the same territory should induce a connected subgraph. Territory compactness is required to guarantee that customers within a territory are relatively close to each other. Compactness and balance with respect to the number of customers are the most important criteria identified by the firm. In the optimization models included in this research, these criteria are considered as objective functions and the remaining criteria are treated as constraints.

There are two ways to address balancing. In this version of the problem, the balance with respect to the number of customers is treated as optimization criterion, and the balance with respect to product demand is treated as constraint. This is motivated by the fact that this criterion is directly related with the number of stops that a vehicle makes during the product distribution. According to the company, the best territory design will be that in which compactness and balancing with respect to the number of customers are reached.

To obtain an optimization model that includes all considerations given by the firm, a bi-objective programming model is proposed. In this model two objective functions are minimized. The first objective (f_1) is related to a dispersion measure, because minimizing dispersion is equivalent to maximizing compactness. The second objective (f_2) is associated to the maximum deviation with respect to the target value ($\mu^{(1)}$) in the number of customers. Minimizing the maximum deviation allows to be closer to the average size of the number of customers. In this work, the objective of the pMP is used as a dispersion measure (f_1).

In a few words, the problem consists of finding a p-partition of V according to the specified planning criteria of balance with respect to the sales volume and connectivity, in such way that both performance measures, dispersion (f_1) and the maximum deviation with respect to the target number of customers on each territory (f_2) are minimized. We assume all parameters are known with certainty.

4.2 Mathematical Formulations

Decision variables

$$x_{ji} = \begin{cases} 1 & \text{if basic unit } j \text{ is assigned to territory with center in } i;\ i, j \in V, \\ 0 & \text{otherwise.} \end{cases}$$

In that sense $x_{ii} = 1$ implies i is a territory center.

Suppose $Q^i = \sum_{j\in V} w_j^{(1)} x_{ji} - \mu^{(1)} x_{ii}$ represents the unbalance with respect to the number of customers in territory with center in i, $i \in V$. So, the relative deviation in territory with center in $i \in V$ is given by

$$\left| \frac{Q^i}{\mu^{(1)}} \right| \tag{4.1}$$

This expression given as an absolute value can be decomposed into a positive ΔW_i^+ and a negative ΔW_i^- part as follows: $\left|\frac{Q^i}{\mu^{(1)}}\right| = \Delta W_i^+ + \Delta W_i^-$, where $\frac{Q^i}{\mu^{(1)}} = \Delta W_i^+ - \Delta W_i^-$, and $\Delta W_i^+ \Delta W_i^- = 0, i \in V$. Based on this, the following bi-objective MILP model is obtained.

4.2.1 Bi-objective Programming Model

(BOTDP)

$$\text{Min } f_1 = \sum_{j\in V}\sum_{i\in V} d_{ji}x_{ji} \tag{4.2}$$

$$\text{Min } f_2 = \max_{i\in V}\{\Delta W_i^+ + \Delta W_i^-\} \tag{4.3}$$

Subject to:

$$\Delta W_i^+ \Delta W_i^- = 0 \qquad i \in V \tag{4.4}$$

$$\Delta W_i^+ - \Delta W_i^- = \frac{\sum_{j\in V} w_j^{(1)} x_{ji} - \mu^{(1)} x_{ii}}{\mu^{(1)}} \qquad i \in V \tag{4.5}$$

$$\sum_{i\in V} x_{ii} = p \tag{4.6}$$

$$\sum_{i\in V} x_{ji} = 1 \qquad j \in V \tag{4.7}$$

$$\sum_{j\in V} w_j^{(2)} x_{ji} \geq (1-\tau^{(2)})\mu^{(2)} x_{ii} \qquad i \in V \tag{4.8}$$

$$\sum_{j\in V} w_j^{(2)} x_{ji} \leq (1+\tau^{(2)})\mu^{(2)} x_{ii} \qquad i \in V \tag{4.9}$$

$$\sum_{j\in \cup_{v\in S}(N^v \setminus S)} x_{ji} - \sum_{j\in S} x_{ji} \geq 1 - \mid S \mid \qquad i \in V, \quad S \subset [V \setminus (N^i \cup \{i\})] \tag{4.10}$$

$$x_{ji} = \{0,1\} \qquad i,j \in V \tag{4.11}$$

$$\Delta W_i^+, \Delta W_i^- \geq 0 \qquad i \in V \tag{4.12}$$

Objective (4.2) represents the dispersion measure. In this sense, minimizing dispersion is equivalent to maximizing compactness. The second objective (4.3) represents the maximum deviation with respect to the target value of number of customers. So, balanced territories should have small deviation with respect to the average number of customers. Constraints (4.4) and (4.5) establish the relationship with the absolute value of $\frac{Q_i}{\mu^{(1)}}$. Constraint (4.6) guarantees the creation of exactly p territories. Constraints (4.7) guarantee that each node j is assigned to only one territory. Constraints (4.8)-(4.9) represent the territory balance with respect to the sales volume as it establishes that the size of each territory must lie within a range (measured by tolerance parameter $\tau^{(2)}$) around the average size. Constraints (4.10) guarantee the connectivity of the territories. Observe that, as usual, there is an exponential number of such constraints.

Note that objective (4.3) is a piece-wise linear function. So, BOTDP can be linearized by replacing (4.3) by

$$\text{Min } f_2 = \gamma \tag{4.13}$$

and introducing constraints given by

$$\gamma \geq \Delta W_i^+ + \Delta W_i^-, \forall i \in V. \tag{4.14}$$

In addition, it can be shown (see Lemma 4.1) that nonlinear constraints (4.4) are not needed.

The resulting bi-objective MILP is called LBOTDP. Model LBOTDP does not include the set of nonlinear constraints $\Delta W_i^+ \Delta W_i^- = 0, i \in V$. It is because, when a feasible solution of LBOTDP is obtained, those $L = \{l : l \in V\}$ in which both ΔW_l^+ and ΔW_l^- take value different to zero can be identify easily. When this happens, it is always possible to get a feasible solution in which at least one of these ΔW_l^+ or ΔW_l^- takes a value equal to zero (see Lemma 4.1) and the new γ value which will be equal or better than the actual γ value is recomputed.

Lemma 4.1. For any feasible solution $(X, \Delta W)$ of LBOTDP such that $\Delta W_l^+ > 0$ and $\Delta W_l^- > 0$ there exists a feasible solution $(\bar{X}, \Delta\bar{W})$ for LBOTDP such that $X = \bar{X}$ and $\Delta\bar{W}_l^+ + \Delta\bar{W}_l^- = 0, l \in V$, where, $f_1(X) = f_1(\bar{X})$ and $f_2(\Delta W) \geq f_2(\Delta\bar{W})$.

Proof. Let $(X, \Delta W)$ be a feasible solution of $LBOTDP$ with corresponding objective function values given by (f_1, f_2). We will focus especially in constraints (4.5) and (4.14). For each $l \in L$ where $L = \{l \in V : \Delta W_l^+ > 0$ and $\Delta W_l^- > 0\}$, there are two cases.

- Suppose $\Delta W_l^+ \geq \Delta W_l^-$. Let $\Delta\bar{W}_l^+ = \Delta W_l^+ - \Delta W_l^-$ and $\Delta\bar{W}_l^- = 0$. Clearly, $\Delta\bar{W}_l^+ - \Delta\bar{W}_l^- = \Delta W_l^+ - \Delta W_l^-$. Then, the new values $\Delta\bar{W}_l^+$ and $\Delta\bar{W}_l^-$ satisfy the constraints (4.14) as well. And $\Delta\bar{W}_l^- \Delta\bar{W}_l^+ = 0$

- Similarly if $\Delta W_l^+ < \Delta W_l^-$. Let $\Delta\bar{W}_l^- = \Delta W_l^- - \Delta W_l^+$ and $\Delta\bar{W}_l^+ = 0$. Again, $(\Delta\bar{W}_l^+, \Delta\bar{W}_l^-)$ is feasible.

Since, $\Delta\bar{W}_i^+ + \Delta\bar{W}_i^- \leq \Delta W_i{}^+ + \Delta W_i^-, \forall i$, it follows that $\bar{X}$ is equal to X and ΔW is less than $\Delta\bar{W}$. It implies that, $f_2(\Delta W) \leq f_2(\Delta\bar{W})$ and the proof is completed. □

From a practical point of view, it has been clearly established that both f_1 and f_2 are in conflict. It has been observed empirically that when attempting to reach the best possible dispersion measure the maximum deviation with respect to the target number of customers increases and viceversa. This justifies the bi-objective model.

4.3 NP-Completeness Proof

The characterization of a given optimization problem is an important issue to be investigated before to decide the application of any solution technique. For a given optimization problem, it is possible to define a closely related *decision problem* which is a question whose answer is “yes” or “no”. The classification of the decision problem is used to determine the optimization problem complexity. An optimization problem is NP-hard if the corresponding decision problem is NP-complete. There are different classes of decision problems, for more information see [57] and [30].

Complexity classes for decision problems can be defined in terms of any mathematical formalism for algorithms, such as the *Turing machine*. *P* is the class of decision problems for which there are deterministic polynomial time algorithms for telling whether the answer is *yes* or *no*. Another complexity class of decision problem is the *NP* class. *NP* is the class of decision problems that can be solved by a deterministic algorithm in an exponential time. So, for a problem to be in *NP* it does not require that every instance can be answered in polynomial time for some deterministic algorithm. It simply required that, if x is an instance of the problem, then there exists a *concise* certificate for x, which can be validated in polynomial time [57]. The following definitions are taken from Papadimitriou and Steiglitz [57].

Definition 4.1. *P* is the class of decision problems that can be solved by a deterministic algorithm in polynomial time.

Definition 4.2. *NP* is the class of decision problems that can be solved in polynomial time by a nondeterministic algorithm.

The class of the *NP-complete* problems has the following very interesting properties (see Papadimitriou and Steiglitz [57] for more details).

- Up to now, no deterministic polynomial time algorithm for an NP-complete problem is known.
- If there is a polynomial algorithm for any NP-complete problem, then there are polynomial algorithms for all NP-complete problems.

According to Garey and Johnson [30], the process of devising an NP-completeness proof for a decision problem Π will consist of the following four steps:

1. showing that Π is in NP and Π has a concise certificate that can be validated in a polynomial time,
2. selecting a known NP-complete problem Π',
3. constructing a transformation Ω from Π' to Π, and
4. proving that Ω is a (polynomial) transformation.

There are alternative techniques used for proving NP-completeness results. Probably, the proof by *restriction* is the most simple, and perhaps the most frequently applied. Basically, an NP-completeness proof by restriction for a given problem $\Pi \in NP$ consists simply of showing that Π contains a known NP-complete problem Π' as a special case. The core of such a proof lies in the specification of the additional restrictions to be placed on the instances of Π so that the resulting restricted problem will be identical to Π'. We do not require that the restricted problem and the known NP-complete problem be *exact* duplicates of one another, but rather that there be an "obvious" one-to-one correspondence between their instances that preserves "yes" or "not" answers. This one-to-one correspondence, which provides the required transformation of Π' to Π, is often so apparent that it needs not even be given explicitly [30].

In the following the NP-hardness proof of the bi-objective territory design problem (given by LBOTDP model) is developed. This proof is carried out by using the characterization of some optimization subproblems that are identified into the bi-objective problem.

NP-Completeness Proofs

SLBOTP$_1$ Subproblem. The first NP-hardness proof is carried out over a single-objective problem identified in the LBOTDP model. Let SLBOTDP$_1$ be the problem that minimizes the dispersion measure given by (4.2) subject to the following constraints: i) creation of a fixed number of territories (4.6), ii) exclusive assignment (4.7), iii) balancing territories with respect to the sales volume ((4.8)-(4.9)), and iv) connectivity in all territories (4.10).

For determining the complexity of the SLBOTDP$_1$ problem, the NP-complete decision problem well-known as *Min-sum Multicenter Problem* is used. In this decision problem, connectivity constraints are not validated, however, since a given solution, a polynomial time is required to verify if the connectivity constraints are satisfied or not. The *Breadth First Search* (BFS) algorithm can be used to make this test and it requires at most $O(n + m)$ time.

Definition 4.3. Min-Sum Multicenter. [30]

Instance: Graph $G = (V, E)$, weight $w(v) \in Z_0^+$ for each $v \in V$, lenght $l(e) \in Z_0^+$ for each $e \in E$, positive integer $K \leq |V|$, positive rational B.

Question: Is there a set P of K "points on G" such that if $d(v)$ is the length of the shortest path from v to the closest point in P, then $\sum_{v \in V} d(v) * w(v) \leq B$?

Theorem 4.1. The optimization problem defined by SLBOTDP$_1$ model is NP-hard.

Proof. Suppose that we have an algorithm that generates a solution X (p-partition of V). Therefore, it is possible to validate if X is a feasible solution of SLBOTDP$_1$ in a polynomial time. This validation is really simple for constraints ((4.8)-(4.9)). Even though there is an exponential number of connectivity constraints (4.10), the validation of these constraints is done in a polynomial time by answering the question: Is there a p-partition of V such that each partition induces a connected subgraph?. Thus, even that the BFS algorithm must be applied p times (once for each partition), the total connectivity analysis requires a polynomial time.

Now, suppose that we have a particular instance of the commercial TDP, where $\tau(2) = \infty$ and G is a complete graph. For this particular case, constraints ((4.8)-(4.9)), and (4.10) are redundant. Therefore, the resulting problem is the NP-hard problem well-know in the literature as the p-median problem [44] which has associated the NP-complete decision problem known as *Min-Sum Multicenter Problem* (see [30]).

Then, the optimization problem defined by SLBOTDP$_1$ is a NP-hard problem. It was showed that SLBOTDP$_1$ can be seem as an special case of a decision problem classified as NP-complete. □

SLBOTP$_2$ Subproblem. The second NP-hardness proof is carried out over a single-objective problem identified in the LBOTDP model. Let SLBOTDP$_2$ be the problem that minimizes the maximum deviation with respect to the number of customers (given by (4.4) and (4.5)) subject to these constraints: i) creation of a fixed number of territories (4.6), ii) exclusive assignment (4.7), iii) balancing territories with respect to the sales volume ((4.8)-(4.9)), and iv) connectivity in all territories (4.10).

For determining the complexity of this problem (SLBOTDP$_2$), a variant of the NP-complete problem well-known as *Cut Into Connected Components of Bounded Size* is used. This variant is called *Cut Into Connected Components of Bounded Weight* and it can be defined as follows,

Definition 4.4. Cut Into Connected Components of Bounded Weight [30]
Instance: Graph $G = (V, E)$, integer bound D, weight $w(v)$ for each $v \in V$.
Question: Is there a partition of V into disjoints sets V_1 and V_2 such that $\sum_{v \in V_1} w(v) \leq D$ and $\sum_{v \in V_1} w(v) \leq D$, and both induces connected subgraphs of G?

Theorem 4.2. The optimization problem defined by SLBOTDP$_2$ model is NP-hard.

Proof. Suppose that we have an algorithm that generates a solution X (p-partition of V). Given a solution X, it is possible to validate if X is a feasible solution of SLBOTDP$_2$ in a polynomial time. Constraints (4.6) and (4.7) are redundant and constraints ((4.8)-(4.9)) are validated easily. The connectivity constraints (4.10) can be verified in a polynomial time, like in the previous proof.

Now, suppose that we have a particular instance of the commercial TDP, where $p = 2$, $w_j^{(1)} = 1, j \in V$, and $\tau(2) = \infty$. Then, the resulting problem consist of finding a 2-partition of V that minimizes the maximum deviation with respect to the number of customers assigned to each partition, subject to both V_1 and V_2 induces a connected subgraph $G' = (V_k, E(V_k)), k = 1, 2$. Observe that solving this problem is equivalent to find a "yes" answer to the NP-complete decision problem known as *Cut Into Connected Components of Bounded Size Problem.* Where each node has assigned a weight equal to 1 and D is greater or equal than the largest partition size of V. This decision problem belongs to the NP-complete class (see [42]). Then, SLBOTDP$_2$ is a NP-hard problem and the proof is completed. □

LBOTDP Complexity According to the previous complexity proofs, the optimization problem for the bi-objective commercial TDP is NP-hard. It becomes from the fact that its single-objective versions are NP-hard.

Theorem 4.3. The optimization problem defined by LBOTDP model is NP-hard.

CHAPTER 5

A MULTIOBJECTIVE EXACT METHOD

Heuristic procedures have been developed as solution methodologies in all previous work on commercial territory design problems and these have been addressed for solving single-objective versions of this problem. The bi-objective version of the problem described in Chapter 4 is new in the literature of TDPs and then, there is not a solution technique for solving this problem.

Multiple techniques for solving multiobjective problems have appeared in the literature [22, 15]. One of the most important techniques used in multiobjective programming is the ε-constraint method [25]. There are certainly other techniques such as the weighted sum scalarization method, for instance. However, the ε-constraint method seem best suited for nonconvex problems such as the problem addressed here. In addition, the single-objective approach (ICGP-TDP) described in Chapter 3 can be efficiently exploited within an ε-constraint framework. So, for solving the bi-objective version of this problem (described in Chapter 4) an integration of the ICGP-TDP and the ε-constraint method is developed. The resulting procedure is called ε-ICG. Two alternatives of this method are implemented: the traditional ε-constraint method (εCM) which guarantees obtaining weakly efficient solutions and a modified version of the ε-constraint method (IεCM) in which slack variables are included to guarantee efficient solutions. The last technique was recently proposed by Ehrgott and Ruzika [25] in their improved ε-constraint method.

This chapter is organized as follows. Section 5.1 presents two optimization models called LBOTDP$_\varepsilon$ and LBOTDP$_\varepsilon^+$ which are based on the εCM and the IεCM, respectively. Section 5.2 describes the solution procedure (called ε-ICGP) and experimental work is discussed in Section 5.3. Finally, some conclusions are given in Section 5.4.

5.1 The ε-Constraint Models

The ε-constraint method is based on a scalarization where one of the objective functions is minimized while all the other objective functions are bounded from above by means of additional constraints [25]. In this implementation of the ε-constraint method, the objective function (4.13) was chosen as the function to be bounded by an ε value. It was done because the firm has defined precisely the range of variation (associated to the maximum deviation γ with respect to the average number of customers) in which a solution is attractive to them. In general, those solutions with relative deviation (γ) less than or equal to 5% are attractive to firm. So, different values around this value can be swept in an easy way. The model

$$
\begin{aligned}
(\text{LBOTDP}_{\varepsilon}) \qquad & \text{Min } f_1 \\
& \text{Subject to:} \\
& \text{(4.5)-(4.12) , (4.14)} \\
& \gamma \leq \varepsilon \qquad (5.1)
\end{aligned}
$$

corresponds to the traditional ε-constraint (εCM) formulation for the LBOTDP model. The objective function f_1 is given explicitly by (4.2) and (4.13) is a lower bound of γ. For more details about (4.5)-(4.12), and (4.14) see Section 4.2.1.

It is well known that the εCM method guarantees the obtaining of weakly efficient solutions that can be efficient solutions. However, when you have an optimal solution to the LBOTDP$_{\varepsilon}$ problem is not easy to verify if this solution is an efficient solution or not. To eliminate this weakness, Ehrgott and Ruzika [25] introduced a modification in the traditional formulation. They incorporate nonnegative slack variables and with this modification the new ε-constraint method (IεCM) guarantees the obtaining of efficient solutions. Let

$$
\begin{aligned}
(\text{LBOTDP}_{\varepsilon}^{+}) \quad & \text{Min } f_1 - \lambda s \qquad (5.2) \\
& \text{Subject to:} \\
& \text{(4.5)-(4.12) , (4.14)} \\
& \gamma + s \leq \varepsilon \qquad (5.3)
\end{aligned}
$$

$$s \geq 0, \tag{5.4}$$

be the modified ε-constraint formulation (IεCM) in the studied problem, where λ is a nonnegative weight.

The slack variables introduced in LBOTDP$_{\varepsilon}^{+}$ provide information about the efficiency of a solution [25]. The main difference between LBOTDP$_{\varepsilon}$ and LBOTDP$_{\varepsilon}^{+}$ is that the ε-constraint in LBOTDP$_{\varepsilon}^{+}$ is always active at optimality.

5.2 Solution Algorithm (ε-ICGP)

The LBOTDP$_{\varepsilon}$ and LBOTDP$_{\varepsilon}^{+}$ formulations allow us to obtain weakly-efficient and efficient fronts, respectively, by using different ε values. For each fixed value of ε a single-objective problem LBOTDP$_{\varepsilon}$ or LBOTDP$_{\varepsilon}^{+}$ is solved. Note that, each of these single-objective problems (LBOTDP$_{\varepsilon}$ and LBOTDP$_{\varepsilon}^{+}$) is NP-hard. In addition, constraints (4.10) can not be written out explicitly as there are an exponential number of them.
The ICGP-TDP procedure was modified in a such way that both LBOTDP$_{\varepsilon}$ and LBOTDP$_{\varepsilon}^{+}$ formulations can be used to solve the bi-objective commercial territory design problem. Specifically, the *GenerateRelaxedModel* (in ICGP-TDP, Section 3.4) was adapted according to the ε-constraint formulations and the resulting procedure is called ε-ICGP.

There are a few multiobjective districting applications with connectivity constraints and these have been addressed through heuristic procedures [62, 2]. To the best my knowledge, there are no references in the literature on multiobjective districting that provide efficient solutions. In this case, it is possible to obtain weakly-efficient and efficient solutions through ε-ICGP using LBOTDP$_{\varepsilon}$ and LBOTDP$_{\varepsilon}^{+}$ formulations. That is, for each ε value the ICGP-TDP procedure is called and it obtains an optimal solution to the problem when it is feasible. The output of the ε-ICGP procedure is a set of efficient solutions that belong to the Pareto front.

The ε-ICGP procedure is described in Algorithm 4. Note that when $\lambda = 0$ is passed as argument to ε-ICGP, the associated solution method is the traditional εCM (see model LBOTDP$_{\varepsilon}$). However, when $\lambda > 0$ then the associated solution method is the IεCM (see model LBOTDP$^+_\varepsilon$). Observe that, when $\lambda = 0$ both formulations LBOTDP$_\varepsilon$ and LBOTDP$^+_\varepsilon$ yield the same optimal solution for each ε value. Algorithm ε-ICGP was coded in C++ and compiled with the Sun C++ 8.0 compiler under Solaris 9 Operating System. The ICGP-TDP procedure (as described in Section 3.4) calls ILOG CPLEX 11.2 on its iterative optimization process.

Algorithm 4 ε-ICGP(λ, ε_0,δ).

```
Input:
  $\lambda$ := Weight parameter.
  $\varepsilon_0$ := Initial $\varepsilon$ value for bounding the objective given by $f_2$
  $\delta$ := step size for computing the next $\varepsilon$ value
Output:  $D^{\text{eff}}$ Efficient solution set
  $D^{\text{eff}} \leftarrow \emptyset$, $\varepsilon \leftarrow \varepsilon_0$
  while ($\varepsilon > 0$) do
    $S \leftarrow$ ICGP-TDP($\lambda, \varepsilon$)
    if (S is optimal) then
      $D^{\text{eff}} \leftarrow D^{\text{eff}} \cup S$
      $\varepsilon \leftarrow \varepsilon - \delta$
    else
      return $D^{eff}$
    end if
  end while
  return $D^{\text{eff}}$
```

While it is true that LBOTDP$^+_\varepsilon$ is more attractive than LBOTDP$_\varepsilon$ as it guarantees efficient solutions, an interesting issue to investigate is the computational effort employed by each model to properly assess the trade-off between solution quality and time.

5.3 Experimental Work

In the experimental work, randomly generated instances based on real-world data provided by the industrial partner were used. Each instance topology was randomly generated as a planar graph. A tolerance $\tau^{(2)} = 0.05$ with respect to sales volume was considered. Three different instance sets defined by $(n, p) \in \{(60, 4), (80, 5), (100, 6)\}$ were used. For each of these sets, 10 different instances were generated. Additionally, another set with five instances for $(150, 6)$ was generated. The time limit for ε-ICGP was set to 4 hours and the step size was $\delta = 0.001$. As it was mention before, solutions with maximum

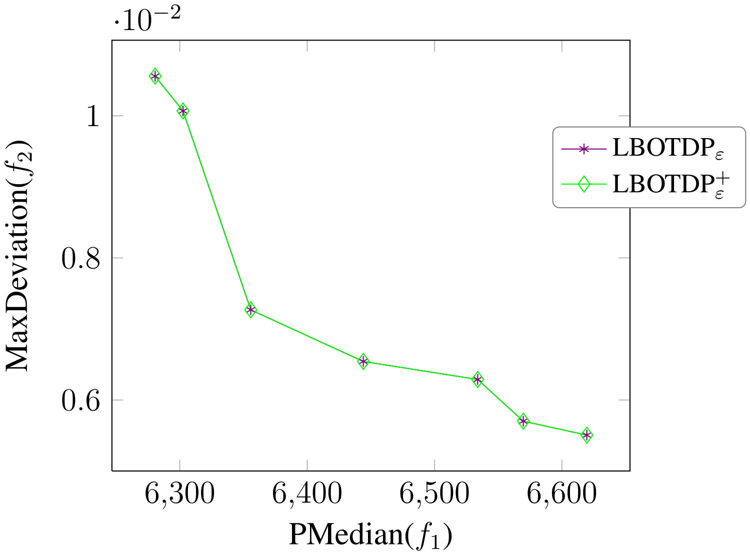

Figure 5.1: Comparison of LBOTDP$_\varepsilon$ and LBOTDP$_\varepsilon^+$ on an instance with 80 BUs and 5 territories.

deviation less than or equal to 5% from the average number of customers are attractive to the firm. Therefore, this value was used as the initial value of ε to bound the objective f_2. The procedure described in Algorithm 4 was used to optimize both the traditional and the improved formulations (LBOTDP$_\varepsilon$ and LBOTDP$_\varepsilon^+$, respectively).

The time required for both LBOTDP$_\varepsilon$ and LBOTDP$_\varepsilon^+$ formulations is first addressed. All instance sets were tested using both formulations. It was observed that there was not a significative difference between these formulations with respect to the time and in most of the cases the set of solutions found through LBOTDP$_\varepsilon$ and LBOTDP$_\varepsilon^+$ optimization was the same. In other words, the stronger structure given by LBOTDP$_\varepsilon^+$ model takes about the same amount of computational effort. Observe that, the optimization process over all instances tested stopped by time limit (4 hours). It is possible to find more efficient points if the time limit increases. So, when the time is unbounded, the optimization process continues until ε reaches the smallest value such that the problem has no feasible solutions.

Figures 5.1, 5.2, and 5.3 show instances where the fronts obtained by the traditional εCM (LBOTDP$_\varepsilon$) and the IεCM (LBOTDP$_\varepsilon^+$) are the same.

Ehrgott and Ruzika [25] show in their work that the traditional ε-constraint method (εCM) (in this case LBOTDP$_\varepsilon$) does not guarantee efficient solutions while the improved

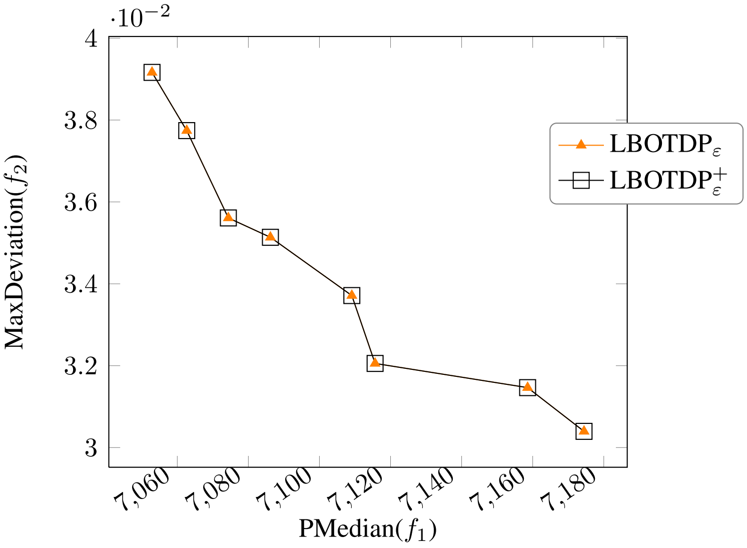

Figure 5.2: Comparison of LBOTDP$_\varepsilon$ and LBOTDP$^+_\varepsilon$ on an instance with 100 BUs and 6 territories.

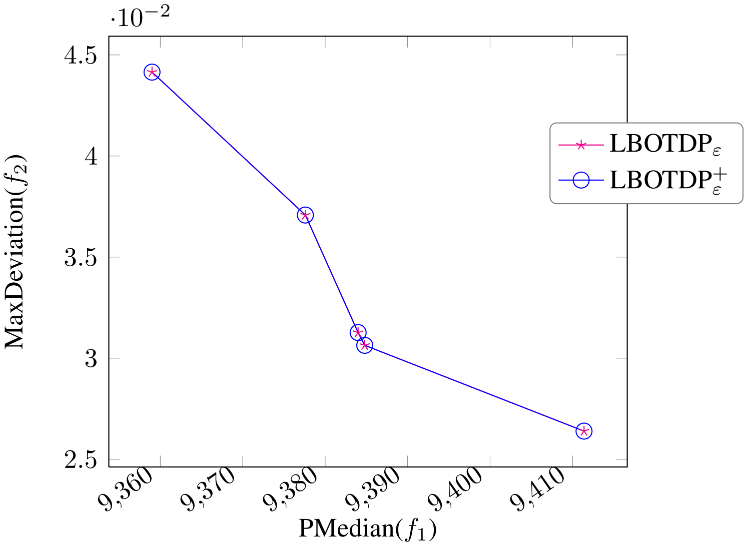

Figure 5.3: Comparison of LBOTDP$_\varepsilon$ and LBOTDP$^+_\varepsilon$ on an instance with 150 BUs and 6 territories.

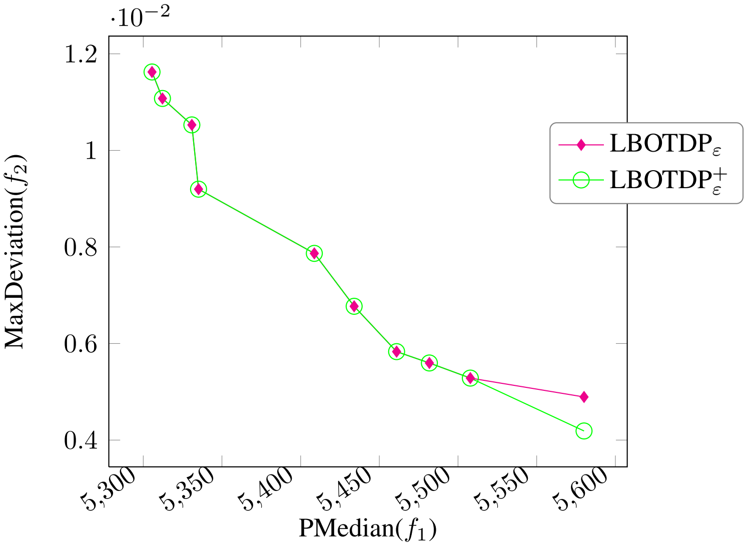

Figure 5.4: A) Comparison of LBOTDP$_\varepsilon$ and LBOTDP$_\varepsilon^+$ on an instance with 60 BUs and 4 territories.

ε-constraint (IεCM) always guarantees this property. This weakness of the traditional ε-constraint is illustrated in two instances belonging to set $(60, 4)$, see Figures 5.4 and 5.5. Figure 5.4 shows us that the fronts reported by LBOTDP$_\varepsilon$ and LBOTDP$_\varepsilon^+$ present a difference when ε is closer to zero. In Figure 5.5 most of the solutions obtained by LBOTDP$_\varepsilon$ optimization are weakly efficient. These weakly efficient solutions (obtained by LBOTDP$_\varepsilon$) are really far from the efficient solutions reported by LBOTDP$_\varepsilon^+$ optimization.

The second part of this experimental work was carried out to analyze two situations that frequently take place in the firm. The first situation occurs when the number of vehicles in the fleet changes. Sometimes, economical resources decrease in a dramatic way such that the firm needs to reduce the number of vehicles (and employees) used for the distribution of the product. As a consequence the firm needs to modify the current territory design. On the other hand, when the firm experiments an expansion, it should make new employee contracts and introduce more vehicles in its fleet. This in turn means that the workload distribution will be affected and a new alignment of territories will be required. These situations were analyzed using the set of instances with 80 BUs and varying the number of territories. Figure 5.6 shows the set of efficient solutions obtained for an in-

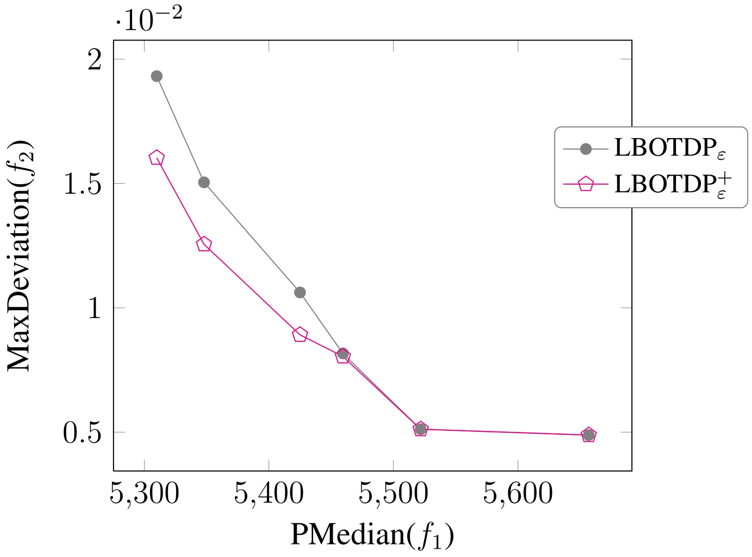

Figure 5.5: B) Comparison of LBOTDP$_\varepsilon$ and LBOTDP$_\varepsilon^+$ on an instance with 60 BUs and 4 territories.

stance with 80 BUs and the number of territories $p \in \{5, 6, 7\}$. Obviously, the dispersion measure (f_2) decreases when the number of territories increases. However, it was observed that when p increases, the unbalance with respect to the number of customers is higher than when p decreases. It is because few combinations of BUs allow to hold the connectivity constraints satisfied on each territory. So, the distribution of workload has more unbalance for large values of p. The decision maker needs to analyze these alternatives. She or he needs to determine what kind of territory design is better for the economical interests to the company. All instances tested with 80 BUs and $p \in \{5, 6, 7\}$ have the same behavior shown in Figure 5.6. The results were obtained using the LBOTDP$_\varepsilon^+$ model, that is, all are efficient solutions.

The second part of this last experiment was carried out to analyze the change in the Pareto front, when the tolerance ($\tau^{(2)}$) changes. We tested the $(60, 4)$ instances for ($\tau^{(2)} \in \{0.05, 0.03, 0.015, 0.01\}$) using LBOTDP$_\varepsilon^+$ model. For instance, Figure 5.7 shows different Pareto frontiers obtained by optimizing the same instance using different $\tau^{(2)}$ values and the stopping rule was set to time limit of 4 hours. We observed the Pareto front for $\tau^{(2)} \in \{0.05, 0.03\}$ is the same. In contrast, the frontier changes when $\tau^{(2)} = 0.015$, observe that some points from the front of $\tau^{(2)} = 0.05$ remain in the frontier for $\tau^{(2)} =$

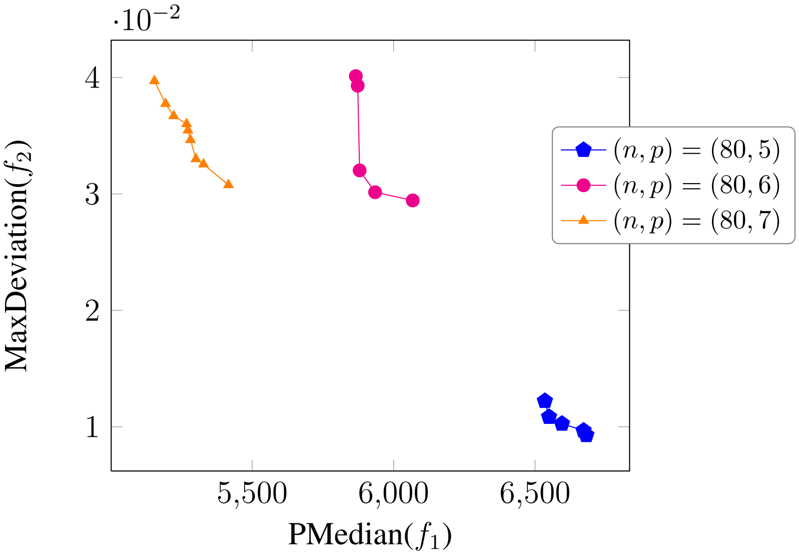

Figure 5.6: Changes in the efficient solutions when p changes.

0.015 and additional efficient solutions are found within the time limit (4 hours).

Pareto front for $\tau^{(2)} = 0.01$ (Figure 5.7) shows the largest change with respect to the other Pareto fronts. Observe for instance, the solution with smallest f_1 (dispersion measure) in this front is really far from the frontiers given by $\tau^{(2)} \in \{0.05, 0.015\}$. This illustrates how the front deteriorates as $\tau^{(2)}$ gets smaller.

5.4 Conclusions

This chapter describes a procedure for solving a bi-objective territory design problem with connectivity and balancing constraints. The solution technique is based on the ε-constraint method and a cut generation procedure.

In the implementation of the exact solution procedure, two variants of the ε-constraint method are developed, i) the traditional method which guarantees the obtaining of weakly efficient solutions, and ii) the first modification proposed by Ehrgott and Ruzika [25] (in the improved ε-constraint method) which guarantees the obtaining of efficient solutions. In the computational work, it was observed there is not significative difference between the time required by both LBOTDP$_\varepsilon$ and LBOTDP$_\varepsilon^+$ models.

The performance of the proposed procedure is evaluated over a set of instances. It

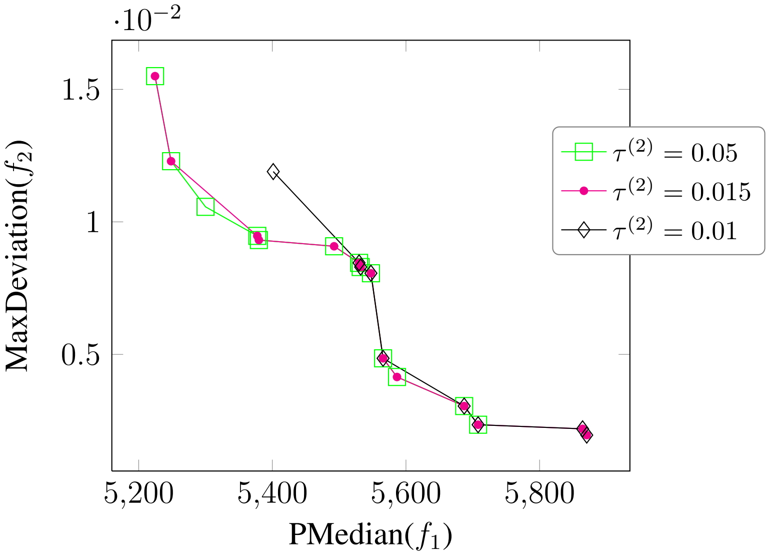

Figure 5.7: Comparison among Pareto fronts for different values of $\tau^{(2)}$.

was observed that instances with up to 150 BUs and 6 territories are solved in a reasonable time. This is a significant result because in the general territory design literature exact solutions have been reported for instances of no more than 50 BUs. Note that this result is for the single objective case. As far as multiobjective territory design with connectivity constraints is concerned, there are no exact methods to the best of my knowledge.

CHAPTER 6

MULTIOBJECTIVE HEURISTIC PROCEDURES

The bi-objective commercial territory design problem belongs to the class of NP-hard problems (see proof in Chapter 4). Experimental work in Chapter 4 shows that large instances of the problem addressed in this work are practically intractable even for the single-objective version (Chapter 3), and it is not possible to obtain efficient solutions in polynomial time. Therefore, the use of heuristic methods is the best alternative for obtaining approximate efficient solutions for relatively large instances.

In general, heuristic methods can be seen as simple procedures that provide satisfactory, but not necessarily optimal, solutions to complex problems in an easy way and in short time. In this work, the well-known framework of Scatter Search (SS) is used to develop a heuristic that allows to obtain approximate efficient solutions to the bi-objective commercial territory design problem.

The proposed procedure (SSMTDP) is a population based metaheuristic that combines aspects of evolutionary algorithms, such as the employment of a combination method in which pairs of solutions are mixed to form several offspring, and strategies to improve these solutions using other heuristic methods. In general, there are five main components in the SS scheme: i) a diversification generation method, ii) an improvement method, iii) a reference set update method, iv) a subset generation method, and v) a solution combination method.

The SSMTDP contains a diversification generation method based on a GRASP framework. Section 6.1 describes different alternatives of GRASP procedures that can be

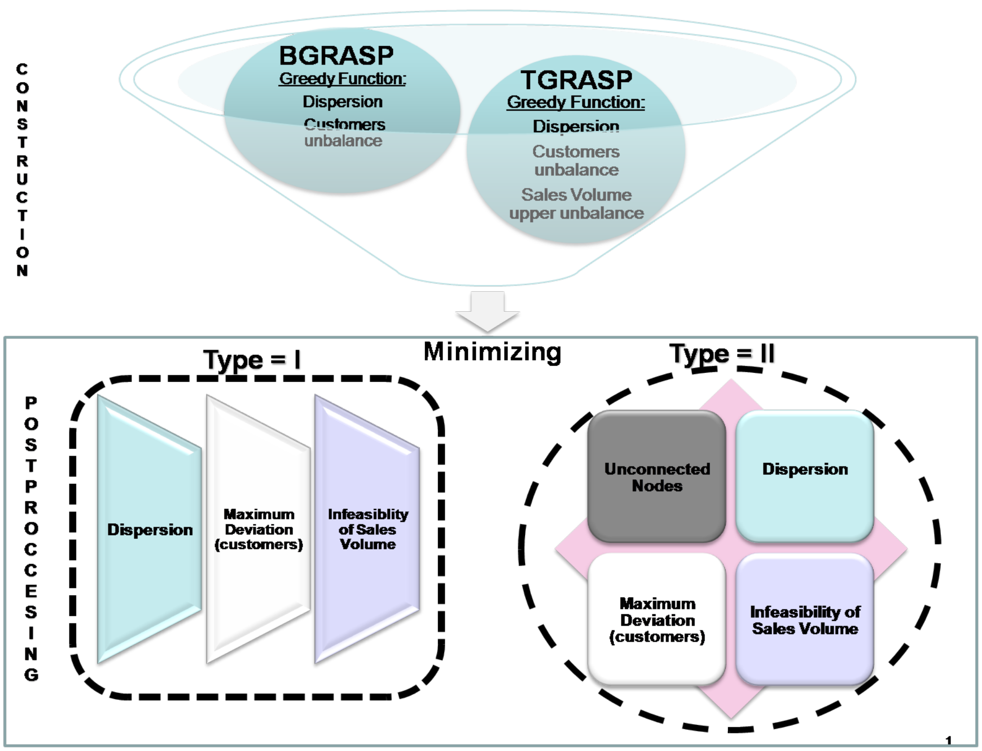

Figure 6.1: General scheme for the proposed GRASP procedures.

used to generate diverse solutions. Finally, Section 6.2 describes the SSMTDP procedure.

6.1 GRASP Procedures

Greedy Randomize Adaptive Search Procedure (GRASP) is a well-known metaheuristic that captures good features of both pure greedy algorithms and random construction procedures [27]. It has been widely used for successfully solving many combinatorial optimization problems. GRASP is an iterative process in which each major iteration consists typically of two phases: construction and post-processing. The construction phase attempts to build a feasible solution and the post-processing phase attempts to improve it. The motivation for GRASP in this application is due to the fact that during the construction phase it is always possible to keep the hard connectivity constraints (4.10), the multiple objectives can be easily evaluated in a merit function, and it is relatively simple to sweep the efficient frontier by using different weights to the multiple objectives for generating diverse solutions.

In this dissertation, two different GRASP schemes called BGRASP and TGRASP are introduced. Each one of them has two variants. For instance, BGRASP-I is a GRASP procedure that uses a merit function based on two components: dispersion and maximum deviation with respect to the target value in the number of customers. This method maintains connectivity as a hard constraint during the construction and post-processing phases. The BGRASP-I post-processing phase consists of optimizing three objective functions: (i) dispersion measure

$$z_1(S) = \sum_{j \in V_t, t \in T} d_{jc(t)}, \tag{6.1}$$

(ii) maximum deviation with respect to the number of customers

$$z_2(S) = \frac{1}{\mu^{(1)}} \max_{t \in T} \left\{ \max\{w^{(1)}(V_t) - \mu^{(1)}, \mu^{(1)} - w^{(1)}(V_t)\} \right\}, \tag{6.2}$$

and (iii) total infeasibility

$$z_3(S) = \frac{1}{\mu^{(2)}} \sum_{t \in T} \max \left\{ w^{(2)}(V_t) - (1 + \tau^{(2)})\mu^{(2)}, (1 - \tau^{(2)})\mu^{(2)} - w^{(2)}(V_t), 0 \right\} \tag{6.3}$$

related to the balancing of sales volume (constraints (4.8) and (4.9)).

In contrast, BGRASP-II does not consider connectivity during the construction phase, its merit function is the same used in BGRASP-I, but during post-processing phase, BGRASP-II adds connectivity as an objective function. So, the goal in its post-processing phase is to minimize four objective functions: dispersion (6.1), maximum deviation 6.2, total infeasibility (6.3), and total number of unconnected BUs. BGRASP-II is proposed because it yields a larger searching space than when the connectivity is kept as a hard constraint (BGRASP-I).

TGRASP-I and TGRASP-II are described in a very similar way as BGRASP-I and BGRASP-II, respectively. The difference is that the merit function in TGRASP-I and TGRASP-II has three components: dispersion, maximum deviation with respect to the number of customers and maximum infeasibility with respect to constraints (4.9). The GRASP strategies are described in a single scheme, see Algorithm 5 and Figure 6.1. In Figure 6.1, observe that, when connectivity is considered as a hard constraint, the post-processing phase is *type = I* and during all GRASP process the assignment of nodes to territories keeps the connectivity requirement. In contrast, when the connectivity is relaxed during the construction phase (BGRASP-II and TGRASP-II), the post-processing phase tries to minimize four objective functions (that is, *type* = *II* in Figure 6.1). The output in any GRASP strategy is an efficient solution set as long as at least one feasible solution is obtained during the GRASP process.

Algorithm 5 shows the general scheme for the proposed GRASP procedures. An instance of the commercial territory design problem, the maximum number of iterations ($iter_{\max}$), the quality parameter (α), the minimum node degree (f) so that a node $i \in V$ can be selected as initial seed, the maximum number of allowed movements (max_{moves}), the number of objectives (*Obj*) to be optimized in the post-processing phase, and the GRASP strategy are the input. In order to explore the objective space in a better way, for each GRASP iteration a set of weights Λ is selected in such a way that $\lambda \in [0,1]$ for $\lambda \in \Lambda$. The two phases are applied for each $\lambda \in \Lambda$. So, for each iteration and each weight $\lambda \in \Lambda$ a construction phase and a local search phase is applied. The construction and the local search applied depend on the strategy chosen. Note that the merit function in BGRASP-I and BGRASP-II uses a weighted combination of the two original objectives. In contrast, in TGRASP-I and TGRASP-II the balancing constraints (4.8)-(4.9) are relaxed and added to the merit function.

Under strategies BGRASP-I and TGRASP-I, after the construction phase stops, the obtained solution may be infeasible with respect to the sales volume. Then, to obtain feasible solutions, infeasibility is treated as an objective to be minimized during the post-processing phase. In these strategies, this phase consists of systematically applying the local search sequentially to each of the three objectives individually. That is, first local search is applied using z_1 as the merit function in a single-objective manner. After a local optimum is found, the local search is continued with z_2 as merit function, and then z_3. Finally, the initial objective z_1 is used after the local optimum is obtained for the last objective. During the search, the set of non-dominated solutions is updated at every solution. It is also clear that the order of this single-objective local search strategy implies different search trajectories, that is, optimizing in the order (z_1, z_2, z_3) generates a trajectory different from (z_2, z_3, z_1), for instance. In BGRASP-II and TGRASP-II strategies, after the construction phase stops, the obtained solution may be infeasible not only with respect to sales volume balance, but with respect to the connectivity constraints as well. At the end of the GRASP strategies, an approximation of the Pareto front is reported.

Algorithm 5 General scheme for BGRASP and TGRASP strategies

```
Input:
  $\alpha$:= GRASP RCL quality parameter
  $iter_{\text{max}}$:= GRASP iterations limit
  $f$:= Minimum node degree in the initial seeds
  $max_{\text{moves}}$:= Maximum number of movements in the post-processing phase
  $Obj$:= Number of objectives to be optimized during the post-processing phase
  $strategy$:= BGRASP-I, BGRASP-II, TGRASP-I or TGRASP-II
Output: $D^{\text{eff}}$: set of efficient solutions
  $D^{\text{eff}} \leftarrow \emptyset$
  $D^{\text{pot}}(S) \leftarrow \emptyset$: set of potential efficients solutions
  if ($strategy \in \{$BGRASP-I,BGRASP-II$\}$) then
    for $(\lambda_1, \lambda_2, \dots, \lambda_r)$ do
      for ($l = 1, 2, \dots, iter_{\text{max}}$) do
        $S \leftarrow$ ConstructSolutionBGRASP($\alpha$, $f$, $\lambda$,$strategy$)
      end for
    end for
  else
    for $(\lambda_1, \lambda_2, \dots, \lambda_r)$ do
      for ($l = 1, 2, \dots, iter_{\text{max}}$) do
        $S \leftarrow$ ConstructSolutionTGRASP($\alpha$, $f$, $\lambda$, $strategy$)
      end for
    end for
  end if
  for ($g = 1, \dots, Obj$) do
    $D^{\text{pot}}(S) \leftarrow$PostProcessing($S$, $max_{\text{moves}}$, $strategy$, $g$, $Obj$)
    UpdateEfficientSolutions($D^{\text{eff}}$, $D^{\text{pot}}(S)$)
  end for
  return $D^{\text{eff}}$
```

6.1.1 BGRASP DESCRIPTION

This strategy follows the generic scheme of GRASP (Algorithm 5). A greedy function (6.4) during construction phase is a convex combination of two components weighted by λ which are related to the original objectives: dispersion measure (4.2) and maximum deviation (4.3). The post-processing phase consists of the successive application of single-objective local search procedures (taking one objective at a time). These main BGRASP components illustrated in Algorithm 5 are detailed as follows.

BGRASP CONSTRUCTION PHASE

In general, the construction phase consists of the assignment of BUs to territories keeping balanced territories with respect to the product demand while seeking good objective function values. Before the assignment process takes place p initial points are selected to open p territories, these points are the basis for the assignment process. Previous work showed that this method is very sensitive to the initial seed selection. For instance, when some seeds are relatively close to each other the growth of some territories stops way before reaching balancing. This implies some territories end up being relatively small. So a better spread of the seeds is needed. In order to obtain best initial seeds, p disperse initial points with high connectivity degree are selected. Then, the construction phase starts by creating a subgraph $G' = (V', E(V'))$ where $i \in V'$ if and only if the degree of i, $d(i) \geq f$, where f is a user-given parameter. The seed selection is made by solving a p-dispersion problem [26] on G'. The p nodes are used as seeds for opening p territories. Let $\{i_1, i_2, ...i_p\}$ be this set of disperse nodes. Then from this set, a partial solution $S = (V_1, V_2, ..., V_p)$ is starting by setting $V_t = \{i_t\}, t \in \{1, 2., ...p\}$.

Then, at a given BGRASP construction iteration (see Algorithm 6) p partial territories are considered and the process attempts to allocate an unassigned node keeping balanced territories with respect to the demand. To do that, this method attempts to make assignments to the smallest territory (considering the demand). If BGRASP-I is the strategy selected by the user, the set of possible assignments is given only for those nodes that permit to preserve the connectivity. On the other hand, if the user selected BGRASP-II, the possible assignments are all those nodes that have not been assigned yet. Let V_{t^*} be the territory with smallest demand, $c(t^*)$ the center of V_{t^*} and $N(V_{t^*})$ the set of currently unassigned nodes that can be assigned to V_{t^*}. If $N(V_{t^*})$ is empty the procedure takes the next smallest territory and proceeds iteratively. The cost of assigning a node j

Algorithm 6 ConstructSolutionBGRASP(α, f, λ, *strategy*)

```
Input:
  $\alpha$:= GRASP RCL quality parameter
  $f$:= Minimum node degree which is required to consider a node as an initial seed to open a new territory
  $\lambda$:= weight used in the greedy function
  strategy:= BGRASP-I or BGRASP-II
Output:  $S = (V_1, ..., V_p)$: Solution, $p$-partition of $V$
  $T = \{1, ...p\}, t \in T$:= Territory index
  $c(t)$:= Center of $V_t$
  Flag(t):= $1$ if a territory $t$ is open, $0$ otherwise
  $B \leftarrow V$; $V_t \leftarrow \emptyset$
  $H \leftarrow \{i \in V : |N^i| \geq f$ {Subgraph of $G$ used to select the initial seeds}
  for all $t \in T$ do Flag(t)$\leftarrow 1$
  Compute $p$ disperse points $\{i_1, ..., i_p\}, i_t \in H$
  for all $t \in T$ do
  $c(t) \leftarrow i_t$; $V_t \leftarrow V_t \cup \{i_t\}$; $B \leftarrow B \setminus \{i_t\}$
  while $(B \neq \emptyset)$ do
    $l \leftarrow \arg \min_{t \in T : Flag(t)=1} \frac{w^{(2)}(V_t)}{\mu^{(2)}}$
    if (strategy = BGRASP-I) then
      $N(l) \leftarrow \bigcup_{i \in V_l} \{j \in N^i$ and $j \in B\}$ {only connected nodes}
    else
      $N(l) \leftarrow B$
    end if
    if $(N(l) \neq \emptyset)$ then
      ComputeGreedyFunction $\phi(j, c(l))$ for all $j \in N(l)$
      $\phi_{\min} \leftarrow \min_{j \in N(l)} \phi(j, c(l))$
      $\phi_{\max} \leftarrow \max_{j \in N(l)} \phi(j, c(l))$
      RCL $\leftarrow \{j \in N(l) : \phi(j, c(l)) \in [\phi_{\min}, \alpha(\phi_{\max} - \phi_{\min})]\}$
      Random selection of $k \in$ RCL
      $V_l \leftarrow V_l \cup \{k\}$; $B \leftarrow B \setminus \{k\}$
      $c(l) \leftarrow \arg \min_{j \in V_l} \sum_{i \in V_l} d_{ji}$ {Update center}
    else
      flag(t)$\leftarrow 0$ {Close territory}
    end if
  end while
  return $S = (V_1, ..., V_p)$
```

to territory V_{t^*} is given by

$$\phi(j,t^*) = \lambda f_{\text{disp}}(j,t^*) + (1-\lambda) f_{\text{dev}}(j,t^*), \tag{6.4}$$

where

$$f_{\text{disp}}(j,t^*) = \frac{1}{d_{\text{max}}} \left(\sum_{i \in V_{t^*} \cup \{j\}} d_{ic(t^*)} \right),$$

$$f_{\text{dev}}(j,t^*) = \frac{1}{\mu^{(1)}} \max \left\{ w^{(1)} \left(V_{t^*} \cup \{j\} \right) - \mu^{(1)}, \mu^{(1)} - w^{(1)} \left(V_{t^*} \cup \{j\} \right) \right\},$$

and the normalization parameter is

$$d_{\text{max}} = \frac{(|V| - p)}{p} \max_{i,j \in V} d_{ij}. \tag{6.5}$$

Observe that this greedy function is a weighted sum of the changes produced in the objective values.

Following the GRASP mechanism a Restricted Candidate List (RCL) is built with the most attractive assignments which are determined by a quality parameter $\alpha \in [0,1]$ (specified by the user). The RCL is computed as follows:

$$\phi_{\min} = \min_{j \in N(t^*)} \phi(j, c(t^*)), \tag{6.6}$$

$$\phi_{\max} = \max_{j \in N(t^*)} \phi(j, c(t^*)), \tag{6.7}$$

$$\text{RCL} = \{ j \in N(t^*) : \phi(j, c(t^*)) \in [\phi_{\min}, \phi_{\min} + \alpha(\phi_{\max} - \phi_{\min})] \}. \tag{6.8}$$

Then, a node i is randomly chosen from the RCL. We update the territory $V_{t^*} = V_{t^*} \cup \{i\}$ and the center $c(t^*)$ is recomputed. This is the adaptive part of GRASP. We proceed iteratively until all nodes are assigned. At the end of the process a p-partition $S = (V_1, V_2, ..., V_p)$ is obtained. This partition may be infeasible with respect to the balance of sales volume. In a few words, the proposed construction procedure tries to build territories similar in size with respect to the demand attribute. The next component of BGRASP is the post-processing or improvement phase.

BGRASP Post-processing Phase

The main idea of this local search is to successively apply a single-objective local search scheme (one objective function at a time) to avoid the cycling behavior observed in multiobjective search. This idea is moticated by its successful application in other MOCO methods [54]. This process starts with the final solution obtained in the construction phase. Then, it starts with a solution $S = \{V_1, ..., V_p\}$. Additionally, for each $V_t \in S$ a center $c(t) \in V_t$ is associated and a territory index $q(i) = t$ is known for $i \in V_t$. S may be infeasible with respect to the balancing constraints (4.8) and (4.9), so in this phase BGRASP attempts to obtain feasible solutions by simultaneously searching for solutions that represent the best compromise between the objective functions. In order to obtain feasible solutions during this phase, balancing constraints (4.8) and (4.9) are dropped and are considered as an additional objective function instead. As mentioned before, in the case of BGRASP-I, there are three objectives that are minimized: (i) dispersion measure (6.1), (ii) maximum deviation with respect to the number of customers (6.2), and (iii) infeasibility related to the balancing of sales volume (6.3). In contrast, the post-processing phase in BGRASP-II adds another minimizing objective to those three objectives used in BGRASP-I. It is given by

$$z_4(S) = \bigcup_{t \in T} |\eta(V_t)|, \tag{6.9}$$

where

$$\eta(V_t) = \bigcup_{r \in \{1,...,q-1\}} B_r^t.$$

The function z_4 computes the total number of unconnected nodes. For territory $B^t = (V_t, E(V_t))$ let $B_r^t = (X_r, E(X_r))$ be the r-th connected component of B^t, for $r = 1, \ldots, q$. For simplicity, let $c(t) \in X_q$. Evidently, if $q = 1$ B^t is connected. Otherwise we have $q - 1$ sets of nodes that do not connect with the center $c(t)$ of territory V_t.

The Post-processing phase attempts to find potential efficient solutions in the neighborhood of S. For doing that, a neighborhood $N(S)$ is defined. This neighborhood is formed by the solution set obtained by all possible moves such that a basic unit $i \in V_{q(i)}$ is reassigned to any adjacent territory $V_{q(j)}, q(j) \neq q(i)$, into the p-partition defined by S. Note that, Algorithm 8 works for any GRASP strategy proposed in this work. Observe that, when the current solution is connected, a movement is allowed only if the resulting solution keeps the connectivity requirement. It means that, when BGRASP-I is used, only connected moves are allowed and when BGRASP-II is used, this condition is activated once a connected solution has been found. Each possible movement *move*(i,j) deletes i from territory $q(i)$ and inserts it into territory $q(j)$, $(i,j) \in E, q(i) \neq q(j)$. For example, suppose we have a partition S with the structure $S = (..., V_{q(i)}..., V_{q(j)}, ...)$, if the *move*($i,j$) is selected, the neighbor solution $\bar{S}$ is given by $\bar{S} = (..., V_{q(i)} \setminus \{i\}, \ldots, V_{q(j)} \cup \{i\}, ...)$. The *move*($i,j$) is accepted only if this improves the value of the objective function that is being optimized in that moment (see Algorithm 8).

Algorithm 7 PostProcessing(S_0, g, Obj)

```
Input:
  $S = S_0$:= Initial solution
  $h = g$:= objective index for starting the linked local search, $g \in \{1, 2, ..., Obj\}$
  $Obj$:= Number of objective functions to be optimized
Output:  $D$: Nondominated solutions set
  Do
  $D \leftarrow \emptyset$, $count \leftarrow 0$
  $N(S)$: {Set of neighbors. In this case set of possible moves}
  A move $(i,j)$ is represented by an arc $(i,j) \in E$ such that $t(i) \neq t(j)$ ie.,
  $N(S) = \{(i,j) \in E$ such that $t(i) \neq t(j)$ under the partition $S\}$
  while ($N(S) \neq \emptyset$) and ($count < iter_{\max}$) do
    $(i,j) \leftarrow$ select_move($N(S)$)
    $N(S) \leftarrow N(S) \setminus \{(i,j)\}$
    acceptable $\leftarrow$ EvaluateMove($S, (i,j), h$)
    if (acceptable) then
      $S_{t(i)} \leftarrow S_{t(i)} \setminus \{i\}$
      $S_{t(i)} \leftarrow S_{t(j)} \cup \{i\}$
      count$\leftarrow$ count$+1$
      Update($N(S)$)
      if (IsFeasible($S$) = YES) then
        UpdateNDS($D, S$)
      end if
    end if
  end while
  if ($h < Obj$) then
    $h = h + 1$
  else
    $h = h - 1$
  end if
  While($h \neq g$)
  return $D$
```

The neighborhood exploration consists of linking single-objective local search evaluations. This is very similar to the local search proposed in MOAMP [9] and used by Molina, Martí, and

Algorithm 8 EvaluateMove($S, (i, j), g$)

```
Input:
  $S$:= Current solution
  $(i, j)$:= Intended move
  $g$:= Objective function index that should be optimized
Output: YES if $(i, j)$ is acceptable NO otherwise
  $\bar{S} \leftarrow S : S_{t(i)} \setminus \{i\}, S_{t(j)} \cup \{i\}$ {new solution from $S$ after move $(i, j)$ is done}
  $\triangle z_g = z_g(S) - z_g(\bar{S})$ {change in the objective value after move $(i, j)$ from $S$}
  if ($z_4(S) = 0$) then
    if ($z_g \neq z_4$) then
      if ($\triangle z_g > 0$) and ($z_4(\bar{S}) = 0$) then
        return YES
      else
        return NO
      end if
    else
      if ($\triangle z_1 || \triangle z_2 || \triangle z_3$) and ($z_4(\bar{S}) = 0$) then
        return YES
      else
        return NO
      end if
    end if
  else
    if ($\triangle z_g > 0$) then
      return YES
    else
      return NO
    end if
  end if
```

Caballero [54]. The linking of single-objective local search schemes is made by considering different ordering of the objective functions being pursued. Assuming the optimization order is chosen as $(z_1(S), z_2(S), z_3(S))$, the local search path is as follows. The first local search starts with S, a solution found in the construction phase, and attempts to find a better solution with respect to the single objective $z_1(S)$ (6.1). Let S^1 be the best solution visited at the end of this search. Then a local search is applied again to find the best possible solution to the problem with the single objective $z_2(S)$ (6.2) using S^1 as initial solution. After that, a local search is applied to find the best solution to the problem considering the single objective $z_3(S)$ (6.3) and S^2 as the initial solution. Finally, a local search using $z_1(S)$ as objective and S^3 as initial solution is done. This phase yields at least three solutions that approximate the best solutions to the single objective problems that result from ignoring all but one objective function. During this phase only feasible solutions are kept and a potential set of nondominated solutions is kept as well (see Algorithm 9). Additionally, efficient solutions may be found because all potential nondominated solutions are checked for inclusion in the efficient set E (see Algorithm 10). This efficient set E is updated according to Pareto efficiency. This check is made over the original objectives: dispersion (6.1) and maximum deviation with respect to the number of customers (6.3) (see Algorithms 9 and 10).

Algorithm 9 UpdateNDS(D, S)

```
Input:
  $D$ Current set of nondominated solutions
  $S$ Candidate solution
Output: $D$ Set of nondominated solutions
  eff $\leftarrow 1$ := 1 if the solution $S$ is efficient, 0 in otherwise
  for $S' \in D$ do
    if $((z_1(S) \geq z_1(S'))$ and $(z_2(S) \geq z_2(S')))$ then
      eff $\leftarrow 0$
    end if
  end for
  if eff then
    for $S' \in D$ do
      if $((z_1(S) \leq z_1(S'))$ and $(z_2(S) \leq z_2(S')))$ then
        $D \leftarrow D \setminus \{S'\}$
      end if
    end for
    $D \leftarrow D \cup \{S\}$
  end if
  return $D$
```

Algorithm 10 UpdateEfficientSolutions(D^{eff}, D^{pot})

```
Input:
  $D^{\text{eff}}$:= Set of current efficient solutions
  $D^{\text{pot}}$:= Set of potential nondominated solutions
Output: $D^{\text{eff}}$: Efficient set
  for $S \in D^{\text{pot}}$ do
    UpdateNDS($D^{\text{eff}}$, $S$)
  end for
  return $D^{\text{eff}}$
```

Definition 6.1. Pareto efficiency. A solution $x^* \in X$ is efficient if there is no other solution $x \in X$ such that $f(x)$ is preferred to $f(x^*)$ according to Pareto order. That is, $x^* \in X$ is efficient if there is no solution $x \in X$ such that $f_i(x) \leq f_i(x^*) \forall i = 1, ..., g$ and at least one $j \in \{1, ..., g\}$ such that $f_j(x) < f_j(x^*)$.

In this case $g = 2$. The linked local search process can be repeated by using a different ordering of the objectives. In this work, different trajectories depending on the number of objectives to be optimized are explored. For instance, in BGRASP the following trajectories were used, these start in the same initial solution: (z_1, z_2, z_3, z_1), (z_2, z_3, z_1, z_2) and (z_3, z_1, z_2, z_3). Each local search stops when the limit of iterations is reached or when the set of possible moves is empty. At the end the output is an approximated Pareto front.

6.1.2 TGRASP Description

TGRASP-I and TGRASP-II are very similar to the BGRASP-I and BGRASP-II, respectively. The main difference is in the construction phase (see Algorithm 11). During this phase the greedy function (6.10) is a convex combination (6.12) of three components: dispersion measure (6.5), maximum deviation (6.5) and maximum infeasibility with respect to the upper bound of sales volume balancing (6.11). The procedure starts with p disperse points (obtained as in BGRASP construction phase) and the cost of assigning a node i to territory t with center $c(t)$ is measured by the greedy function

$$\gamma(j, t) = \lambda_1 f_{\text{disp}}(j, t) + \lambda_2 f_{\text{dev}}(j, t) + \lambda_3 f_{\text{infeas}}(j, t), \tag{6.10}$$

where

$$f_{\text{infeas}}(j, t) = \frac{1}{\mu^{(2)}} \max \Big\{ (1 + \tau^{(2)})\mu^{(2)} - w^{(2)} \left(V_t \cup \{j\} \right), 0 \Big\}, \tag{6.11}$$

Algorithm 11 ConstructSolutionTGRASP(α, f, λ, *strategy*)

```
Input:
  $\alpha$:= GRASP RCL quality parameter
  $f$:= Minimum node degree which is required to consider a node as an initial seed to open a new territory
  $\lambda$:= weight used in the greedy function
  strategy:= TGRASP-I or TGRASP-II
Output:  $S = (V_1, ..., V_p)$: Solution, $p$-partition of $V$
  $T = \{1, ...p\}, t \in T$:= Territory index
  $c(t)$:= Center of $V_t$
  Flag(t):= $1$ if a territory $t$ is open, $0$ otherwise
  $B \leftarrow V$; $V_t \leftarrow \emptyset$
  $H \leftarrow \{i \in V : |N^i| \geq f$ {Subgraph of $G$ used to select the initial seeds}
  Compute $p$ disperse points $\{i_1, ..., i_p\}, i_t \in H$
  for all $t \in T$ do $c(t) \leftarrow i_t$; $V_t \leftarrow V_t \cup \{i_t\}$; $B \leftarrow B \setminus \{i_t\}$
  while (($B \neq \emptyset$) do
    $l \leftarrow \arg\min_{t \in T} \dfrac{w^{(2)}(V_t)}{\mu^{(2)}}$
    if (strategy = TGRASP-I) then
      $N(l) \leftarrow \bigcup_{i \in V_l} \{j \in N^i$ and $j \in B\}$ {only connected nodes}
    else
      $N(l) \leftarrow \cup\{j \in B\}$
    end if
    if ($N(l) \neq \emptyset$) then
      ComputeGreedyFunction $\gamma(j, c(l))$ for all $j \in N(l)$
      $\gamma_{\text{min}} \leftarrow \min_{j \in N(l)} \gamma(j, c(l))$
      $\gamma_{\text{max}} \leftarrow \max_{j \in N(l)} \gamma(j, c(l))$
      RCL $\leftarrow \{j \in N(l) : \gamma(j, c(l)) \in [\gamma_{\text{min}}, \alpha(\gamma_{\text{max}} - \gamma_{\text{min}})]\}$
      Random selection of $k \in$ RCL
      $V_l \leftarrow V_l \cup \{k\}$
      $B \leftarrow B \setminus \{k\}$
      $c(l) \leftarrow \arg\min_{j \in V_l} \sum_{i \in V_l} d_{ji}$ {Update center}
    else
      flag(t)$\leftarrow 0${Close territory}
    end if
  end while
  return $S = (V_1, ..., V_p)$
```

$$\lambda_1 + \lambda_2 + \lambda_3 = 1. \tag{6.12}$$

Note that (6.11) penalizes for violations of the balancing constraint (3.5) only. The post-processing phase of TGRASP procedures is the same as in BGRASP strategies (see Algorithm 7). Note that, in the TGRASP-I and TGRASP-II, during the local search, four objectives are minimized: (i) dispersion measure (6.1), (ii) maximum deviation with respect to the number of customers (6.2), (iii) infeasibility related with balancing of sales volume (6.3), and (iv) total number of unconnected nodes (6.9). The updating of efficient solutions is made considering only feasible solutions.

6.2 The SSMTDP Procedure

The evolutionary approach called Scatter Search (SS) was first introduced in [32] as a meta-heuristic for integer programming. It is based on diversifying the search through the solution space. It operates on a set of solutions, named the reference set (PR), formed by good and diverse solutions of the main population (P). These solutions are combined with the aim of generating new solutions with better fitness, while maintaining diversity. Furthermore, an improvement phase using local search is applied. As detailed in [51], the basic structure of SS is formed by five main methods. A general frame of SS is illustrated in Figure 6.2. SS is a very flexible technique, since some modules of its structure can be defined according to the specific problem. For instance, the diversification and the combination methods are commonly tailored to the specific problem.

The components of the proposed SSMTDP procedure are described as follows:

- A *diversification generation method* that generates a set of initial solutions. It is based on the proposed GRASP procedures described is Section 6.1.
- An *improvement method* that transforms a trial solution into one or more trial solutions. This method is an implementation of the relinked local search (GRASP post-processing) and it is applied to each solution obtained by either the diversification generation or the combination method.
- A *reference set update method* that maintains a portion of the best solutions of the reference set. In this case, the reference set is formed by efficient solutions according to the Pareto sense. When an efficient solution is found, this enters the reference set and those solutions

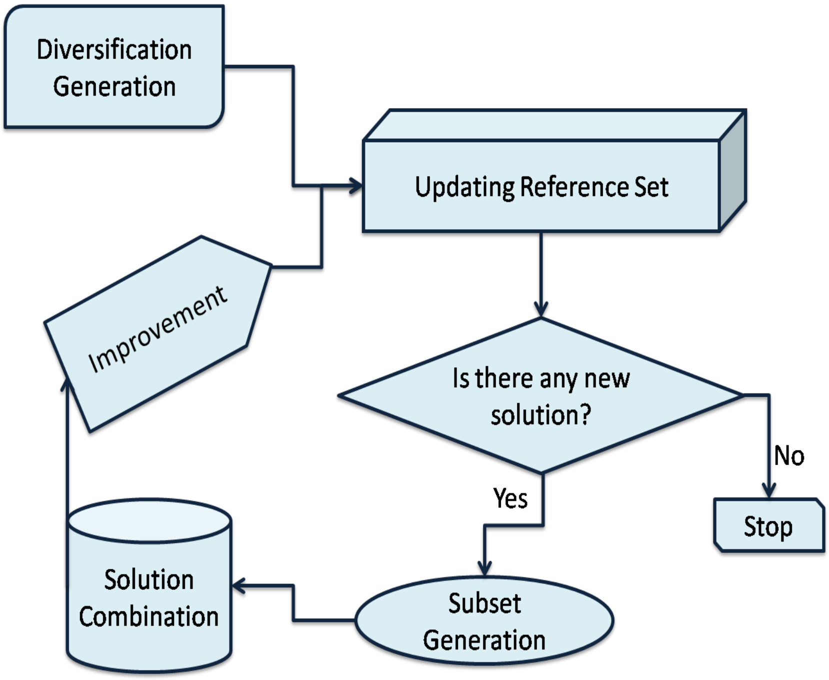

Figure 6.2: General scheme for Scatter Search metaheuristic.

that are dominated by the added solution are deleted from the reference set.

- A *subset generation method* that operates in the reference set in such a way so as to select some solutions to be combined. All possible pairs of solutions from the reference set are selected. During each SSMTDP iteration, a temporal memory is used to avoid those combinations that were done in the previous iteration. In other words, for a specific iteration, the combination process is applied just to those pairs of solutions that were not combined in the previous iteration.

- A *solution combination method* that transforms the solutions built by the subset generation method into one or more combined solution. In this work, three solutions are generated (see Algorithm 12) since each pair of solutions. There are many ways of combining a pair of solutions. In the proposed SSMTDP procedure, this component is developed by attempting to keep good features present in the current solutions. Then, given a pair of solutions S_1 and S_2, these are combined by identifying the best matching between territories. An exhaustive evaluation of the possible matchings requires a high computational effort. Therefore, the method attempts to the best territory matching based on their corresponding territory centers

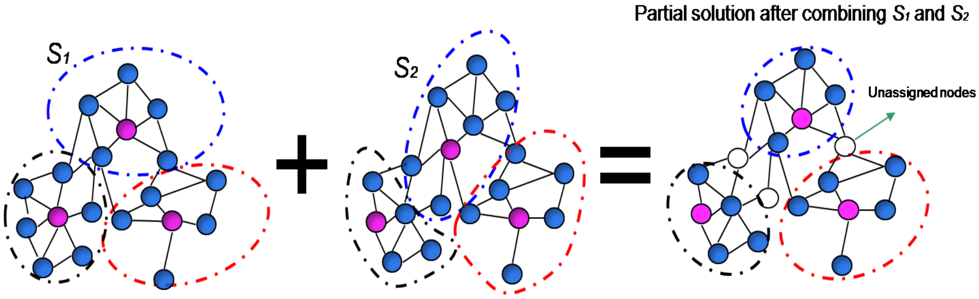

Figure 6.3: Combination of territories between a pair of solutions.

only. This is done by solving an associated assignment problem. The assignment problem used in this method minimizes the sum of distances between the territory centers identified on these solutions. For instance, suppose that solutions S_1 and S_2, with corresponding center sets C_1 and C_2, are to be combined. The assignment problem is solved between the center sets C_1 and C_2, and after that, the resulting assignment is used to determine which territories are matched (see Algorithm 13). Each matching pair (i, j) of this assignment yields a territory in the combined solution by assigning to this territory all those nodes that are common to both, territory with center in i in S_1 and territory with center in j in S_2 (see Algorithm 14). Let S^p be the partial territory design obtained this way. Finally, this partial solution S^p is used as a starting solution for generating three different trajectories, each of them guided by a different objective function, namely: i) dispersion, ii) maximum deviation with respect to the number of customers, and iii) total infeasibility with respect to the sales volume (see Algorithm 15). This, of course, generates three solutions called S_{z_1}, S_{z_2}, and S_{z_3}, respectively.

The SSMTDP stops by iteration limit or by convergence, that is when the reference set does not change (see Algorithm 12). Observe that, the updating of the reference set takes place after a potential set of nondominated solutions is obtained by applying the improvement method over all trial solutions ($S(z_1), S(z_2)$, and $S(z_3)$) generated by the combination method. This strategy was adopted given that the computational effort increases considerably when the typical strategy (i.e., updating after each new feasible solution is generated) is performed.

Figure 6.3 illustrates the process of generating a partial solution by combining a pair of trial solutions S_1 and S_2. In this figure, the pink nodes represent the territory centers. Suppose that after solving the assignment problem, the resulting assignment is represented by territories enclosed by

Algorithm 12 General scheme of the SSMTDP procedure

```
Output:  REFSet Set of efficient solutions (reference set)
  flag$\leftarrow 1$ := 1 if the solution REFSet changes, 0 in otherwise
  $iter = 0, NS = \emptyset, E = \emptyset$, REFSet$= \emptyset$
  REFSet $\leftarrow$ DiverseSolutions $\{$use any of the proposed GRASP procedures$\}$
  while ((flag) and (iter $<$ max$_{\text{iter}}$)) do
    COM$\leftarrow$ SubsetGeneration(REFSet) $\{$pairs of solutions to be combined$\}$
    NS$\leftarrow \emptyset$
    for $(S_1, S_2) \in$ COM do
      $(S_{z_1}, S_{z_2}, S_{z_3}) \leftarrow$ CombinationMethod($S_1, S_2$)
      NS $\leftarrow$ NS$\cup\{S_{z_1}, S_{z_2}, S_{z_3}\}$
    end for
    $E \leftarrow$ Improvement(NS) $\{$calls to post-processing phase of the GRASP procedures by
    minimizing of $(z_1, z_2, z_3, z_4)$ $\}$
    flag$\leftarrow$ UpdateRefSet($E$,REFSet)
    iter$+1$
  end while
  return  REFSet
```

Algorithm 13 CombinationMethod(S_1, S_2)

```
Input:
  $(S_1, S_2)$:= Pair of solutions to be combined
Output:  $(S_{z_1}, S_{z_2}, S_{z_3})$ Three new solutions obtained by combining $S_1$ and $S_2$
  $C_1 \leftarrow \bigcup_{t\in\{1,...,p\}} c(t) \in V_t$, $v_t \in S_1$
  $C_2 \leftarrow \bigcup_{t\in\{1,...,p\}} c(t) \in V_t$, $v_t \in S_2$
  $M \leftarrow \emptyset$ := Matching $\{(i_1, j_1), (i_2, j_2), ..., (i_p, j_p)\}$ between elements from $C_1$ and $C_2$,
  where $i_t \in C_1$ and $j_t \in C_2, t \in \{1, ..., p\}$
  $M \leftarrow$ SolveAssignmentProblem($C_1$,$C_2$)
  $S^p \leftarrow$ BuildPartialSolution($S_1, S_2, M$)
  Compute $I' \subset V$ such that $I'$ contains those nodes that have not been assigned in the
  partial solution $S^p$
  $(S_{z_1}, S_{z_2}, S_{z_3}) \leftarrow$ GenerateNewSolutions($S^p, I'$)
  return  $(S_{z_1}, S_{z_2}, S_{z_3})$
```

Algorithm 14 BuildPartialSolution($(S_1, S_2), M$)

```
Input:
  $(S1, S_2)$:= Pair of solutions to be combined
  $M$:= Matching $\{(i_1, j_1), (i_2, j_2), ..., (i_p, j_p)\}$ between territory centers from $S_1$ and $S_2$
Output: ($S^p = (V_1, V_2, ..., V_p)$) Partial assignment of nodes to territories
  for ($t = 1, ..., p$) do
    $V_t \leftarrow \emptyset$
    $V_t \leftarrow \cup V_{i_t} \cap V_{j_t}$, where $V_{i_t} \in S_1$ and $V_{j_t} \in S_2$
    if ($V_t == \emptyset$) then
      $V_t \leftarrow j_t$
    end if
  end for
  return ($S^p = (V_1, V_2, ..., V_p)$)
```

Algorithm 15 GenerateNewSolutions(S^p, I')

```
Input:
  $S^p$ := Partial solution
  $I'$ := Unassigned nodes
Output: ($S_{z_1}, S_{z_2}, S_{z_3}$) Three new solutions obtained since $S^p$
  $S_{z_1} \leftarrow$ AssignmentGRASP($S^p, I', z_1$) {Merit function for minimizing $z_1(S)$}
  $S_{z_2} \leftarrow$ AssignmentGRASP($S^p, I', z_2$) {Merit function for minimizing $z_2(S)$}
  $S_{z_3} \leftarrow$ AssignmentGRASP($S^p, I', z_3$) {Merit function for minimizing $z_3(S)$}
  return ($S_{z_1}, S_{z_2}, S_{z_3}$)
```

dotted lines of the same color in S_1 and S_2. The partial solution is the basis for generating three new solutions S_{z_1}, S_{z_2}, and S_{z_3}. These new solutions are obtained by adding the unassigned nodes to the partial territories, carrying out three independent GRASP processes. That is, for generating S_{z_1}, the unassigned nodes are assigned to the partial territories through a GRASP process whose merit function is given by the dispersion measure. Then, for generating S_{z_2} the merit function is given by the maximum deviation with respect to the number of customers. Finally, for generating S_{z_3} a merit function that computes the total infeasibility with respect to the balancing of sales volume is used.

When all trial solutions are generated (i.e., when all pairs of solutions are combined), this set of solutions is improved by using the post-processing phase (relinked local search) described in the previous section. At the end, the improvement process reports a potential set of nondominated solutions that can be included in the current reference set. So, each solution from the potential set enters to the reference set if it is an efficient solution with respect to the current set of solutions belonging to reference set. Those solutions that are dominated by the new solution are removed from the current reference set. The SSMTDP stops when there are no new solutions included in the reference set.

CHAPTER 7

EMPIRICAL EVALUATION OF HEURISTICS

In this chapter, the performance of the proposed heuristic procedures is assessed. To this end, two different instances sets of randomly generated instances based on real-world data provided by the industrial partner are tested. The first set is formed by 10 instances with 500 BUs and 20 territories. The second set is formed by 10 instances with 1000 BUs and 50 territories. Additionally, a case study with a real instance with 1999 BUs and 50 territories is included.

This Chapter is organized as follows. Section 7.1 contains experimental work for the proposed GRASP procedures. Section 7.2 shows computational results obtained by applying the proposed SS method (called SSMTDP). A comparison of the proposed GRASP procedures with a generic genetic algorithm (called NSGA-II) is included in Section 7.3, and Section 7.4 shows a comparison of the SSMTDP procedure with a generic state-of-the-art MOCO algorithm known as SSPMO. Section 7.5 shows the results when TGRASP-I is applied to a real-world case. Finally, Section 7.6 summarizes the conclusions of this chapter.

7.1 EVALUATION OF GRASP HEURISTICS

An evaluation of the diverse GRASP strategies described in Chapter 6 is carried out. In all instances, a tolerance parameter $\tau^{(2)} = 0.05$ was used. The input parameters for the GRASP procedures were $f = 2, \alpha = 0.04, \Lambda = \{0, 0.01, 0.02, ...1.0\}$, the total number of GRASP iterations was set to 2020 and the move limit was set to 2000. Experimental work showed that the largest computational effort is during the post-processing phase. The multiple trajectories and the linked local search on each trajectory increase the computational time dramatically. In order to find a good balance between construction and post-processing time, a filtering of solutions was done in order to apply the post-processing phase only over a set of promising solutions which were evaluated according to the merit function given by

$$\rho(S) = \frac{2 f_{\text{disp}}(S)}{(|V| - p) d_{\text{Max}}} + \frac{f^{(1)}_{\text{Tdev}}}{p}, \tag{7.1}$$

where

$$f_{\text{disp}}(S) = \sum_{t \in T} \sum_{j \in V_t} d_{jc(t)}, \tag{7.2}$$

$$f^{(1)}_{\text{Tdev}} = \sum_{t \in T} \left\{ \frac{1}{\mu^{(1)}} \max \left\{ \mu^{(1)} + w^{(1)}\left(V_t\right), \mu^{(1)} - w^{(1)}\left(V_t\right), 0 \right\} \right\}, \tag{7.3}$$

and

$$d_{\text{Max}} = \max_{i,j \in V} d_{ij}. \tag{7.4}$$

The merit function (7.1) is an integration of two components, (7.2) and (7.3). These components are related to the dispersion measure and the maximum deviation with respect to the average number of customers, respectively. Each component of (7.1) is normalized to avoid any bias toward any objective function. The filtering of solutions was carried out by selecting 100 (out of 2020) solutions in such a way that these solutions have the smallest values in the merit function (7.1). Then, the post-processing phase (described in Algorithm 7) was applied over the set of these filtered solutions.

Experimental work was carried out based on a factorial design with two factors (called *strategy* and $type$, respectively). For each factor, two levels were considered, namely,

$$\textit{strategy} \in \{\text{BGRASP, TGRASP}\} \text{ and } \textit{type} \in \{\text{I, II}\},$$

and for each combination of factors, 10 replicates were tested. Figures 7.1 and 7.2 show the efficient frontiers obtained by all GRASP procedures tested over one instance on each size tested ((500,20) and (1000,50), respectively). Observe that, TGRASP-II gives the best and the worst frontier in Figure 7.1 and Figure 7.2, respectively. Thus, for each set, an important issue to investigate is to determine the combination of factors that provides best non-dominated fronts over all instances tested and for different performance measures (the Number of points, k-distance, Size of space covered (SSC), and Coverage of two sets (C(A,B))). These are described in Appendix A. The main goal in the first part of the experimental work was to analyze the effects produced by

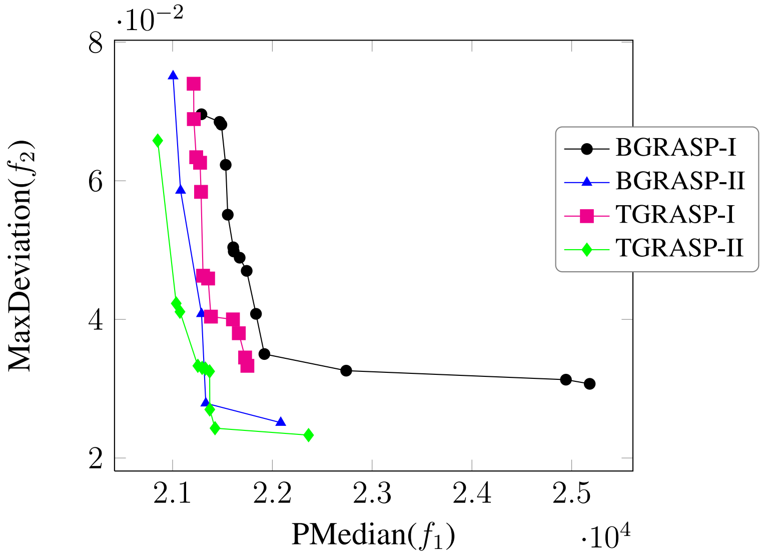

Figure 7.1: Efficient frontiers for an instance with 500 BUs and 20 territories.

each factor over a set of standard performance measures used in multiobjective optimization (see Chapter 2).

Tables 7.1 and 7.2 contain a summary of different performance measures for instances from (500,20) and (1000,50), respectively. An ANOVA for each performance measure was carried out and it was based on a general linear model with interaction of factors. In the results where the ANOVA showed variability from individual factors or from the interaction of these, a residual analysis was carried out to verify the model adequacy. Table 7.3 contains a summary of P-values related to estimated effects over different performance measures. Suppose that a significance level $\alpha = 0.05$ is used during the significance testing. The P-values for instance set (1000,50) show the SSC measure is very sensitive to any change in the individual factors and in any interaction between these. In contrast, when the performance measures are the k-distance(mean) or number of points, it was observed there is not any significant effect produced by individual factors or by interaction of them. Thus, any strategy BGRASP-I, BGRASP-II, TGRASP-I and TGRASP-II is a good alternative according to this performance measure. In the case of k-distance(max) measure, only the factor *type* is statistically significant. That is, the way of handling connectivity affects this performance measure. Recall that, *type* = I means that the connectivity is treated as a hard constraint during construction and post-processing phases, and *type* = II means that during the construction phase the connectivity is not taken into account and it is added as an objective function

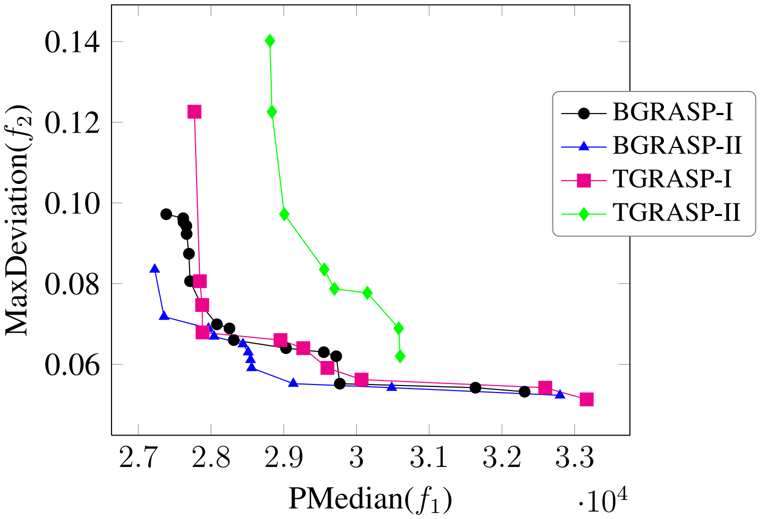

Figure 7.2: Efficient frontiers for an instance with 1000 BUs and 50 territories.

in the post-processing phase. In a similar way, for instances of (500,20), the SSC measure is affected only by the factor *type*. The rest of the performance measures did not present variations by any change in the individual factors and by its interaction. Therefore, in the ANOVA analyses SSC is the only performance measure that presents significant variation generated from the factor levels.

Table 7.1: Summary of metrics used during ANOVA, instances (500,20).

GRASP	k-distance(mean)			k-distance(max)		
procedures	Min	Ave	Max	Min	Ave	Max
BGRASP-I	0.169	0.367	0.729	0.528	0.760	0.995
BGRASP-II	0.145	0.314	0.851	0.307	0.635	0.996
TGRASP-I	0.179	0.308	0.594	0.301	0.586	1.019
TGRASP-II	0.178	0.307	0.485	0.322	0.581	0.900
GRASP	SSC			N. of points		
procedures	Min	Ave	Max	Min	Ave	Max
BGRASP-I	0.642	0.745	0.883	6.000	11.200	17.000
BGRASP-II	0.786	0.845	0.899	5.000	11.500	16.000
TGRASP-I	0.703	0.757	0.853	6.000	11.300	16.000
TGRASP-II	0.638	0.851	0.944	6.000	9.800	16.000

Checking the model adequacy for the performance measure that presented significant variability (SSC), a graphical analysis of residuals was carried out. Figures 7.3 and 7.4 show the residual plots for SSC in the instances set (500,20) and (1000,50), respectively. Note that the residuals

Table 7.2: Summary of metrics used during ANOVA, instances (1000,50).

GRASP	k-distance(mean)			k-distance(max)		
procedures	Min	Ave	Max	Min	Ave	Max
BGRASP-I	0.173	0.352	0.926	0.457	0.689	1.093
BGRASP-II	0.172	0.290	0.437	0.385	0.552	0.727
TGRASP-I	0.156	0.281	0.410	0.376	0.643	0.873
TGRASP-II	0.086	0.262	0.409	0.398	0.526	0.708
GRASP	SSC			N. of points		
procedures	Min	Ave	Max	Min	Ave	Max
BGRASP-I	0.542	0.762	0.867	5.000	12.100	18.000
BGRASP-II	0.622	0.801	0.954	5.000	10.700	18.000
TGRASP-I	0.612	0.737	0.867	4.000	11.300	17.000
TGRASP-II	0.127	0.302	0.548	7.000	11.700	25.000

Table 7.3: Summary of P-values associated to estimated effects from factors to performance measures.

Instances set $(1000, 50)$				
Term	k-distance(mean)	k-distance(max)	N. of points	SCC
strategy	0.287	0.493	0.941	0.000
type	0.379	0.018	0.710	0.000
strategy·type	0.640	0.843	0.504	0.000
Instances set (500,20)				
Term	k-distance(mean)	k-distance(max)	N. of points	SCC
strategy	0.516	0.069	0.466	0.662
type	0.605	0.292	0.584	0.000
strategy·type	0.606	0.331	0.412	0.891

plots show that errors are normally and independently distributed with mean zero, and constant but unknown variance σ^2. The histogram and probability plots show the residuals have the normally property. The plot of residuals versus the order of data shows that the residuals are independent. Note that there is no correlation between the residuals. Finally, the assumption of constant and unknown variance is shown in the plot of residuals versus fitted values. Thus, the general linear model is adequate to these analyses.

Figures 7.5 and 7.6 show the mean values of factor interaction for the SSC measure. Recall that high values of SSC are better than small values. Then, for instances from set (500,20) TGRASP-I and TGRASP-II are slightly better than BGRASP-I and BGRASP-II, respectively (see Figure 7.5). In contrast, for instances from (1000,50) (see Figure 7.6) TGRASP-II obtained the

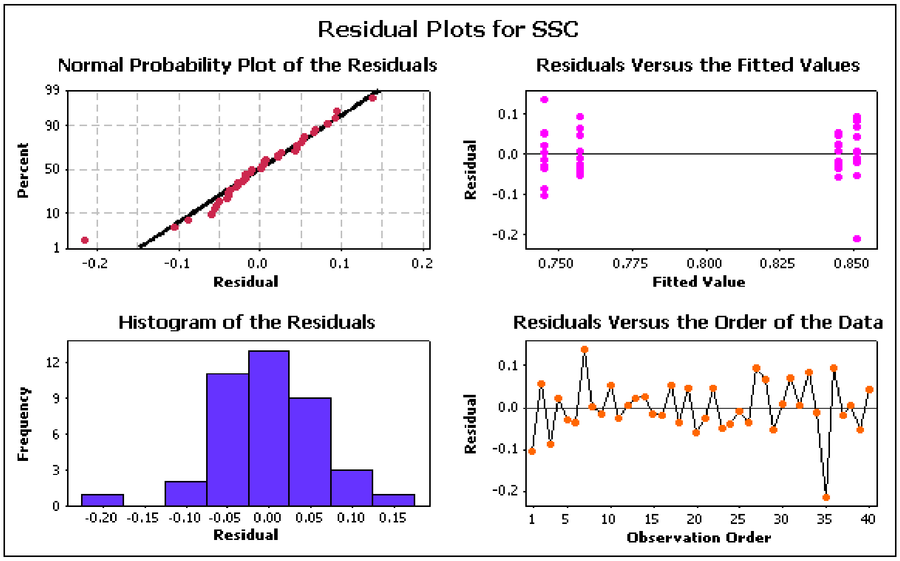

Figure 7.3: Residual plots for the SSC measure in set $(500, 20)$.

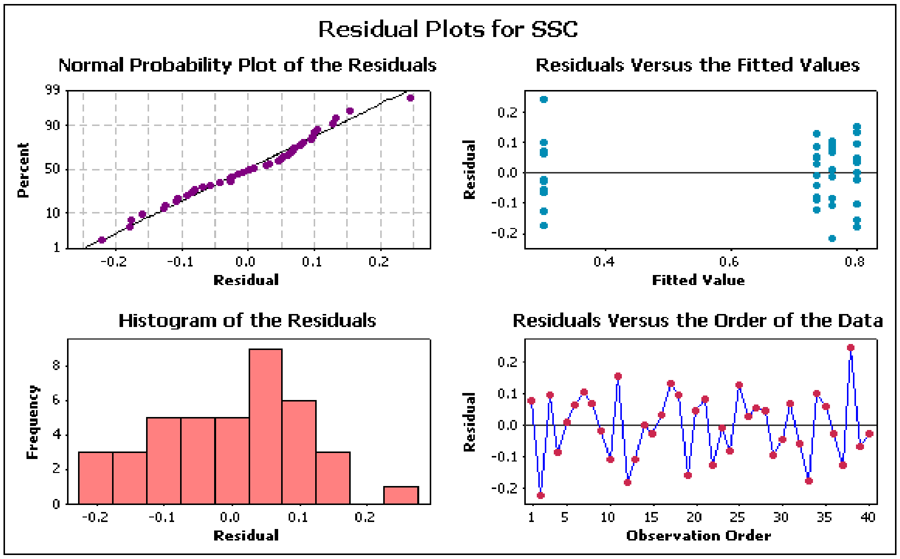

Figure 7.4: Residual plots for the SSC measure in set $(1000, 50)$.

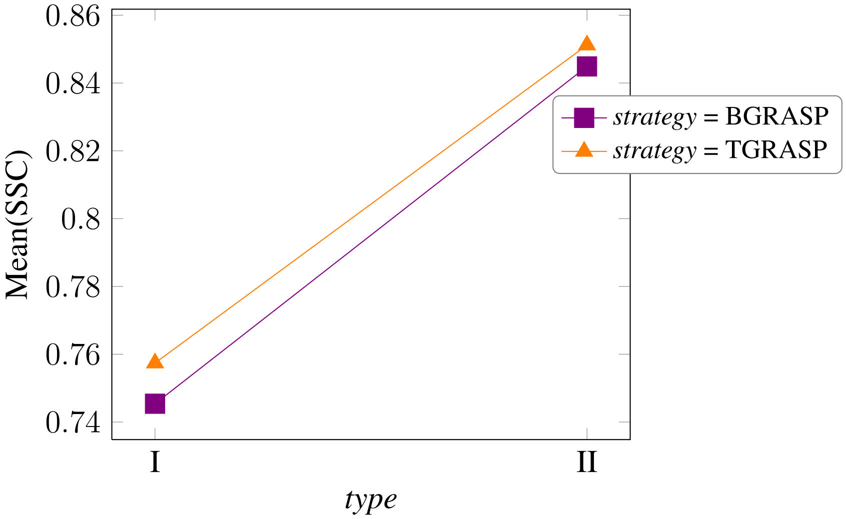

Figure 7.5: Interaction plot for the SSC measure in set (500,20).

worst SSC mean value and it is so far from the SSC mean values given by both BGRASP procedures.

A summary for the coverage of two sets measure is shown in Tables 7.4 and 7.5. Each column on these tables contains the mean proportion of points that are dominated by the procedure indicated by the row label. In Table 7.4, for instance, the values of the third row (BGRASP-II) means that the non-dominated points generated by BGRASP-II dominate 74.1% of those non-dominated points obtained by BGRASP-I and 77.1% of those non-dominated points generated by TGRASP-I. In addition, Table 7.5 shows that for instances from (1000,50) the non-dominated solutions obtained by BGRASP-II tends to dominate 99.1% of those non-dominated points generated by TGRASP-II. In all instances tested, BGRASP-II procedure obtained the best mean values for this performance measure.

Table 7.4: Mean value of coverage of two sets measure for instances from (500,20).

Dominance	BGRASP-I	BGRASP-II	TGRASP-I	TGRASP-II
BGRASP-I	0.000	0.130	0.415	0.366
BGRASP-II	0.741	0.000	0.771	0.486
TGRASP-I	0.486	0.155	0.000	0.303
TGRASP-II	0.651	0.442	0.707	0.000

The last important issue to be evaluated is the optimization time required for each GRASP

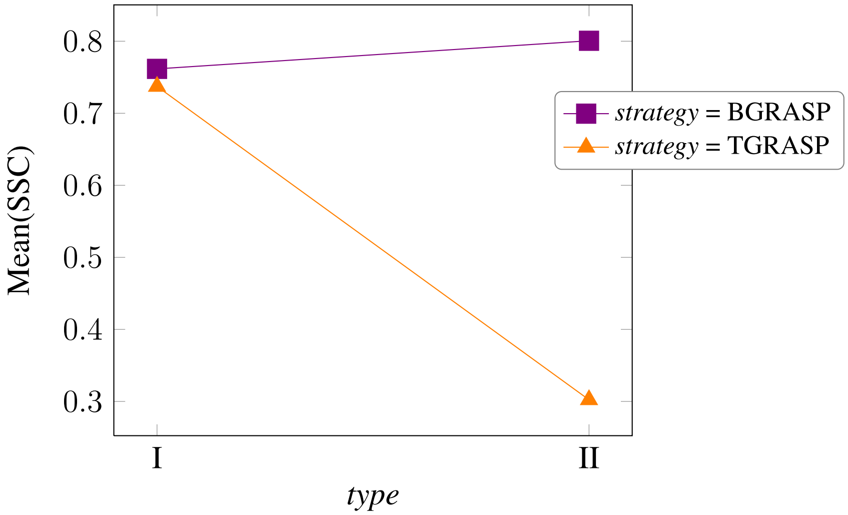

Figure 7.6: Interaction plot for the SSC measure in set (1000,50).

Table 7.5: Mean value of coverage of two sets measure for instances from (1000,50).

Dominance	BGRASP-I	BGRASP-II	TGRASP-I	TGRASP-II
BGRASP-I	0.000	0.328	0.569	0.991
BGRASP-II	0.545	0.000	0.610	0.991
TGRASP-I	0.337	0.304	0.000	0.991
TGRASP-II	0.000	0.000	0.000	0.000

procedure. For instances from (500,20) (see Figure 7.6), the best time is for those procedures that keep the connectivity requirement as a hard constraint during all GRASP process (BGRASP-I and TGRASP-I). In the case of instances from (1000,50) the best time value is obtained by BGRASP strategies, that is, the assignment to the smaller territory during the construction phase is better than the assignment based on a greedy function with three components (6.10).

Table 7.6: Time (seconds) for instances from (500,20).

Procedure	BGRASP-I	BGRAP-II	TGRASP-I	TGRASP-II
Min	6447.20	16419.59	8197.28	14325.75
Average	6587.03	16758.26	8469.09	14435.63
Max	6764.33	17044.63	8763.36	14540.19

Note that BGRASP-I obtains acceptable solutions in a short time (see Tables 7.6 and 7.7). It motivates the incorporation of BGRASP-I as a diversification method into the SSMTDP proce-

Table 7.7: Time (seconds) for instances from (1000,50).

Procedure	BGRASP-I	BGRAP-II	TGRASP-I	TGRASP-II
Min	5345.23	11908.13	16013.41	13098.64
Average	5516.08	12283.92	18610.36	16833.98
Max	5875.26	12736.50	21209.83	18402.87

dure previously described in Chapter 6. The improvement method consists of a linked local search (used in the proposed GRASP strategies), such that four objectives are minimized: i) dispersion, ii) maximum deviation with respect to the number of customers, iii) total infeasibility, and iv) total unconnected nodes. The rest of the components of SSMTDP were described in the previous chapter. The following section contains a summary of the results obtained by applying the SSMTDP procedure over the instance sets.

7.2 SSMTDP RESULTS

The goal of this part of the experimental work was to analyze the SSMTDP performance. The SSMTDP was applied over two instance sets with $(n, p) \in \{(500, 20),(1000, 50)\}$. These instances were tested by applying the proposed GRASP procedures in the previous section as well. The SSMTDP has two stopping criteria, iteration limit and convergence. In this experiment, the maximum number of iterations was set to 10.

During the experimental work, it was observed that in all instances tested, the SSMTDP converged without reaching the iteration limit. It means that in all cases the SSMTDP stopped when there were not new solutions to be added to the reference set. Figure 7.7 shows the behavior exhibited by an instance with 500 BUs and 20 territories. The first front (called BGRASP-I) is the initial solution set generated by the diversification method (BGRASP-I). The following fronts show the solutions that belong to the reference set on each SSMTDP iteration. Recall that the SSMTDP starts with an efficient solution set that is obtained by the diversification method. These solutions are assigned to the initial reference set. After that, each pair of solutions in the reference set is combined to generate three different solutions. The new generated solutions are improved through the linked local search and then, the updating of the reference set is done for obtaining a new reference set. When the reference set does not change, the SSMTDP stops. In the case illustrated in Figure 7.7, the SSMTDP converged in iteration 9. That is, in this iteration, the combination of

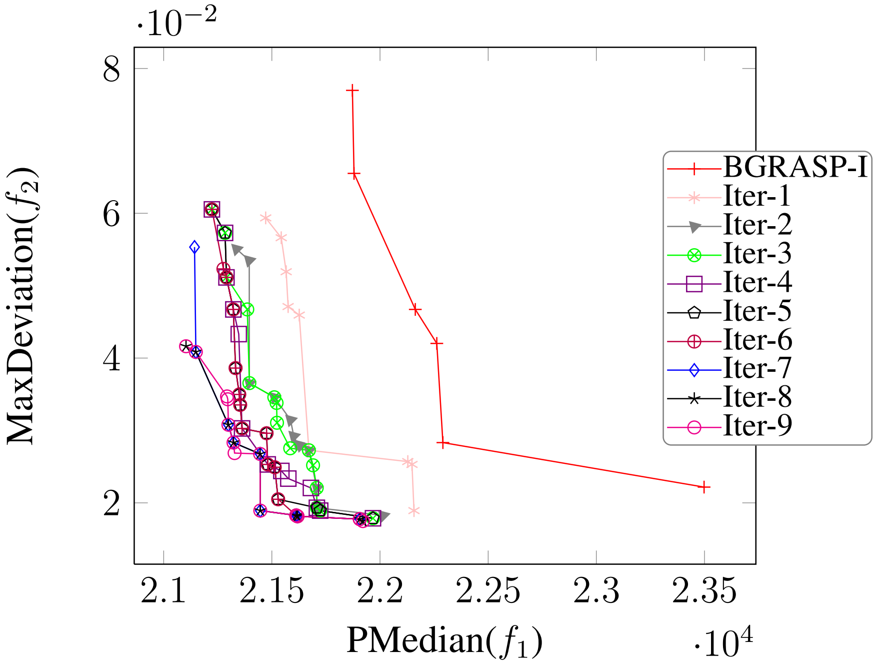

Figure 7.7: SSMTDP performance for an instance with 500 BUs and 20 territories.

solutions from the reference set did not yield potential nondominated solutions to be added to the reference set. Thus, the SSMTDP reports as efficient solutions set those solutions belonging to the reference set in the last iteration.

To illustrate the behavior of SSMTDP by using instances from (1000, 50), Figure 7.8 shows the SSMTDP iterations over an instance with 1000 BUs and 50 territories. In this case the SSMTDP stopped in iteration 8. That is, the combination and improvement procedures did not obtain potential efficient solutions to be added to the reference set. In summary, the efficient fronts obtained by SSMTDP represent a significant improvement with respect to the initial fronts provided by BGRASP-I. It was observed that in all instances tested (20 instances), the SSMTDP method stopped by convergence. These results are used in Section 7.4 for comparing SSMTDP with another SS heuristic called SSPMO.

7.3 Comparing GRASP and NSGA-II

The Non-dominated Sorting Genetic Algorithm (NSGA-II) is a successful evolutionary algorithm that has been applied to many multiobjective problems in the literature [17]. It has been empirically shown this method finds significantly better spread of solutions and better convergence near the true

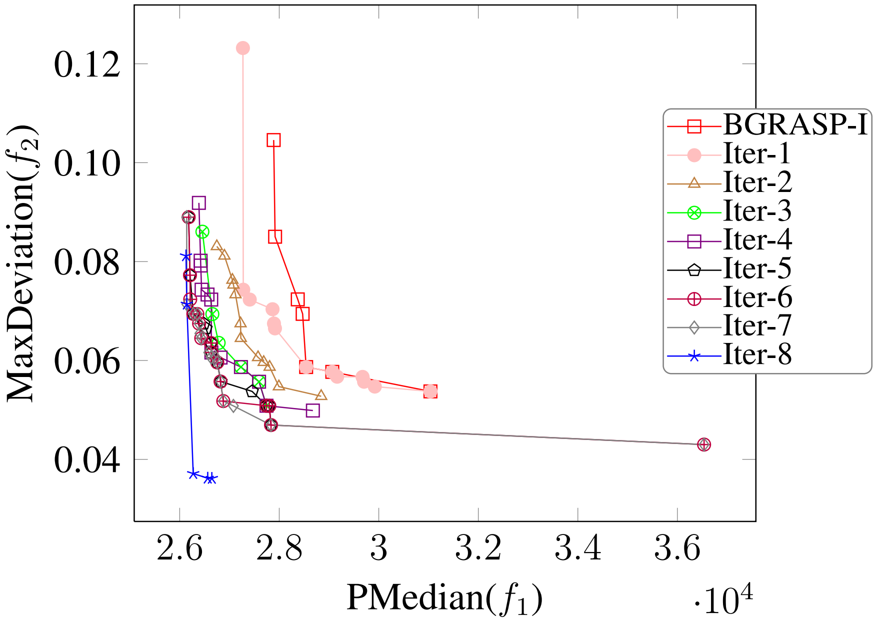

Figure 7.8: SSMTDP performance for an instance with 1000 BUs and 50 territories.

Pareto-optimal front compared to Pareto-archived evolutionary strategy (PAES) [45] and Strength Pareto EA (SPEA). The general description of the NSGA-II procedure is given by Deb et al. [18].

In this dissertation the NSGA-II was adapted to the problem. Four objective functions are minimized: i) dispersion (6.1), ii) maximum deviation with respect to the average number of customers (6.2), iii) total infeasibility with respect to the balancing constraints of sales volume (6.3), and iv) total number of unconnected nodes (6.9). The main features present in this adaptation of the NSGA-II procedure are the following. The generation of solutions consists of randomly selecting p seeds from the set of nodes (V) and assigning the remaining $n - p$ nodes to the closest center. NSGA-II uses different nondomination levels (ranks). In a few words, for each solution h two entities are calculated: 1) domination count d_h which corresponds to the number of solutions that dominate the solution h, and 2) a set of solutions D_h that the solution h dominates. All solutions in the first nondominated front will have their domination count as zero. Then, for each solution h with d_h = 0, each member (g) from S_p is visited, and its domination count is reduced by one. In doing so, if for any member g the domination count becomes zero, it is put in a separate list $\bar{Q}$. These members belong to the second front. Now, the above procedure is continued with each member of $\bar{Q}$ and the third front is identified. The process continues until all fronts are identified.

In the first iteration, the population is sorted based on the nondomination. Then, the fitness function is defined according to the nondomination level. At first, the binary tournament selection is used to create an offspring population $\bar{Q_0}$ of size N. Since elitism is introduced by comparing the current population with previously found best-nondominated solutions, the procedure is different after the initial generation. In the following iterations, the selection is based on the crowded operator which combines the rank (nondomination level) and crowded distance. For more details see [18].

For each pair of solutions two new solutions are obtained. Each new solution copies each center from the father or from the mother with the same probability and the assignment process is equal than the initial generation. For each generated solution, an integer random number is generated in the range [0,4]. If the random number is equal to 0, then the mutation process is not applied. On the other hand, the mutation process takes place by using the kind of move determined by the generated number. The different neighborhoods are defined by the following moves:

1. Select a center and change it for another node randomly selected. Re-assignment of nodes using the new configuration of centers.

2. Select a node in the border of a territory and assign this node to the adjacent territory (keeping connectivity).

3. Select a territory r and a node from an adjacent territory is randomly selected and assigned to r.

4. Interchange two nodes between a pair of territories by holding connectivity.

When the convergence criterion is reached, the best nondominated solutions are filtered to obtain those feasible solutions that are efficient with respect to the dispersion measure and the maximum deviation with respect to the average number of customers.

The NSGA-II was applied over the two instance sets used in the previous section. The number of generations and the population size was set to 500, respectively. On each generation 250 solutions were combined. NSGA-II reported efficient solutions only for a single instance with 500 BUs and 20 territories. For the other 19 instances tested NSGA-II did not obtain feasible solutions and the GRASP strategies reported efficient solutions over all instances tested. It was observed how the method fails on appropriately handling the connectivity constraints. Most of the solutions generated by NSGA-II are highly infeasible with respect to the connectivity constraints, even though the NSGA-II considers this requirement as objective to be minimized. The selection mechanism

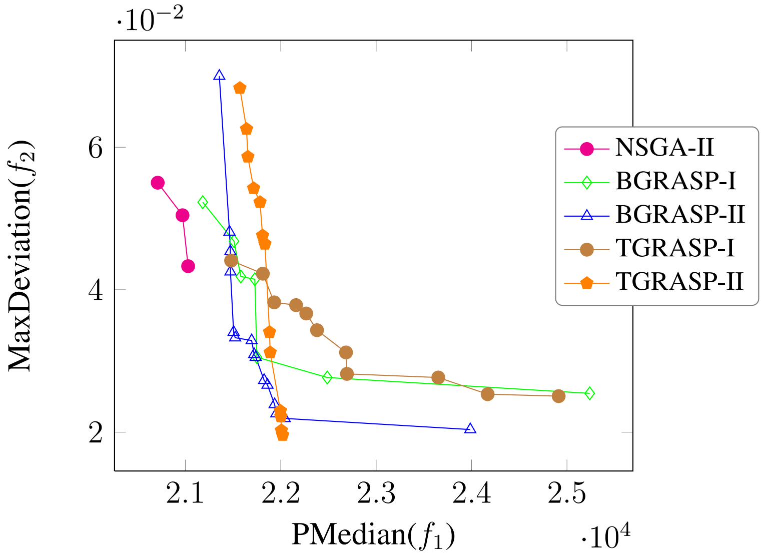

Figure 7.9: Comparison of Pareto fronts for an instance with 500 BUs and 20 territories, GRASP strategies vs. NSGA-II.

and the combining processes are not enough to efficiently handling these very difficult constraints. In contrast, the proposed GRASP procedures are specifically designed to take the connectivity into account when constructing each solution. The other constraints are not as difficult and can be satisfied in the post-processing phase. That is, for this problem exploiting problem structure definitely pays off. Figure 7.9 shows the comparison among the proposed GRASP strategies and the NSGA-II method. Observe that, few efficient solutions from the GRASP strategies are dominated by the efficient set reported by NSGA-II. In addition, all GRASP strategies have efficient points in a region that is not covered by the Pareto front obtained by NSGA-II.

Table 7.8: Summary of metrics for an instance from (500,20).

Procedure	No. Points	k-distance(mean)	k-distance(max)	SSC
BGRASP-I	7	0.412384	0.870831	0.687756
BGRASP-II	15	0.181105	0.713948	0.779688
TGRASP-I	11	0.219406	0.507138	0.642409
TGRASP-II	13	0.210309	0.321244	0.750969
NSGA-II	5	0.203844	0.242975	0.511952

Table 7.8 shows a comparison of the heuristics with respect to different metrics. As can be seen, the results confirm the superiority of any of the GRASP strategies over NSGA-II. In the k-distance(mean), BGRASP-II reports the best value and NSGA-II reached the best value for

Table 7.9: Coverage of two sets C(A,B) for an instance from (500,20).

C(A,B)	BGRASP-I	BGRASP-II	TGRASP-I	TGRASP-II
NSGA-II	0.285714	0.2	0.0909091	0.538462

k-distance(max). The coverage of two sets measure C(A,B) is shown in Table 7.9. The points reported by all GRASP strategies did not dominated any point reported by NSGA-II. In constrast, the points obtained by NSGA-II dominated some points obtained by the GRASP strategies. For example, Table 7.9 shows that NSGA-II dominates 28% of the points obtained by BGRASP-I and 53% of the points reported by TGRASP-II. However, the NSGA-II reported feasible solutions just for a single instance out of 20 instances tested, while GRASP strategies reported feasible solutions for all instances tested. In summary, GRASP strategies outperform the NSGA-II evolutionary algorithm.

In the following section, a comparison between the proposed SSMTDP and SSPMP, a state-of-the-art SS heuristic is done.

7.4 The SSPMO Procedure

SSPMO is a metaheuristic search procedure introduced by Molina et al. [54] initially developed for solving non-linear multiobjective optimization problems; however, it has been adapted for multiobjective clustering problems as well. It consists of a scatter/tabu search hybrid procedure that includes two different phases: i) generation of an initial set of efficient points through various tabu searches (MOAMP), and ii) combination of solutions, and updating of efficient set via scatter search.

The generation of the initial set is based on the MOAMP method proposed by Caballero et al. [9]. To build the initial set of efficient points, MOAMP carries out a series of linked tabu searches (linked means that the last point of one search becomes the initial point of the next search) where each point visited could be included in the final efficient set. The second phase of the MOAMP consists of an intensification search around the initial set of efficient points. For more details see [9] and [54].

The SSPMO procedure creates a reference set (E) using the efficient solutions reported by MOAMP. A list of solutions that have been selected as reference point is kept to prevent the selection of those solutions in future iterations. Then, each solution that is added to the set E, is added to a *TE* (tabu set). A linear-combination method is used to combine reference solutions. All pair of solutions in E are combined and each combination yields four new trial solutions. Each new solution is subject to an improvement method based on MOAMP. Solutions generated after the improvement procedure are tested for possible inclusion in E.

Once all pairs of solutions in E are combined and the new trial solutions are improved, the SSPMO procedure updates the reference set E to follow the next iteration. The first step in the updating process is to choose the best solutions according to each of the objective functions taken separately. In this selection, those solutions belonging to *TE* are not considered. The remaining solutions are chosen by using a metric L_∞, that is a generalization of the Euclidean distance. For each $x \in E \backslash TE$ the minimum distance ($L_\infty^{\text{min}}(x)$) from x to *TE* is computed, and a uniform random number is generated. If it is less than ($L_\infty^{\text{min}}(x)$), then x is declared eligible. Let y be the maximum number of solutions to be combined. Then, $y - g$ solutions with largest minimum distance to *TE* are selected sequentially. Note that, *TE* is updated after each selection in order to avoid choosing points that are too close to each other. The updating process continues until the mean value of ($L_\infty^{\text{min}}(x)$) for the set of eligible solutions falls below a pre-specified threshold mean-distance. For a complete description of SSPMO method, see [54].

The SSPMO method was adapted to the multiobjective commercial territory design problem, the general structure of SSMO is kept. Four objective functions are minimized: i) dispersion (6.1), ii)maximum deviation with respect to the average number of customers (6.2), iii) total infeasibility with respect to the balancing constraints of sales volume (6.3), and iv) total number of unconnected nodes (6.9). The initial solutions set to feed the MOAMP is generated by choosing p seeds (configuration of centers) and each of the remaining BUs is assigned to its closest center. The maximum number of updates of the reference set was set to 10 (equal to the number of iterations used in SSMTDP), the maximum number of tabu solutions was set to 55, the threshold value was set to 0.05, and the maximum number of efficient solutions included in the reference set was set to 100. The neighborhoods are the same that those defined in the NSGA-II method (previous section). For each pair of solutions, 4 new trial solutions are generated.

At the end, the efficient solutions reported by SSPMO are filtered using only those feasible solutions that are efficient with respect to the dispersion measure and the maximum deviation with respect to the average number of customers.

7.4.1 Comparing SSPMO and SSMTDP

In this part of the computational work, the SSMTDP procedure is compared with the SSPMO procedure. Both SS-based procedures stop by convergence or by iteration limit (10 updates of the reference set). Figures 7.10 and 7.11 show a comparison between Pareto fronts obtained by the SSPMO and the SSMTDP procedure, respectively. These results correspond to 10 instances with 500 BUs and 20 territories. Observe that two approximated fronts obtained by the SSMTDP procedure are included, these fronts were obtained by making a variation of the number of moves allowed during the improvement phase. In some cases, the change in the number of moves yields better fronts and in other cases there are not significant changes in the fronts. Clearly, SSMTDP outperforms SSPMO procedure over all instances tested.

In addition, 10 instances with 1000 BUs and 50 territories were tested by applying both SSPMO and SSMTDP procedures. However, using the same stopping criteria as in the previous cases, the SSPMO spent more than 30 days without getting convergence for the first instance tested. Then, the stopping criteria was changed and the iteration limit was set to 2. The SSMTDP converged and reported efficient solutions for all instances tested. The maximum number of moves for this cases was set to 2000. Due that the tremendous computational effort required by the SSPMO procedure, it was not applied over all instances with 1000 Bus and 50 territories. Figure 7.12 shows the performance of the SSPMO and SSMTDP procedures. The approximated front reported by SSPMO corresponds to those solutions that belong to the reference set after iteration 2.

7.5 Case Study

A real-world instance with 1999 Bus and 50 territories was solved by applying the TGRASP-I procedure. The maximum number of moves in the post-processing phase was set to 3000, and the quality parameter in the RCL was $\alpha = 0.05$, the rest of the input parameters are the same as in Section 7.1. Figure 7.13 shows the approximated front reported by TGRASP-I. It is important to mention that the firm solves the territory design problem around twice a year, that is, the company faces the territory design problem every six months. Moreover, the planning department has tried to generate a territory design plan by minimizing a single-objective function, specifically the dispersion measure. For the instance tested in this dissertation, they did not obtain feasible solutions even for the single-objective problem. In contrast, the TGRASP-I procedure reported approximate

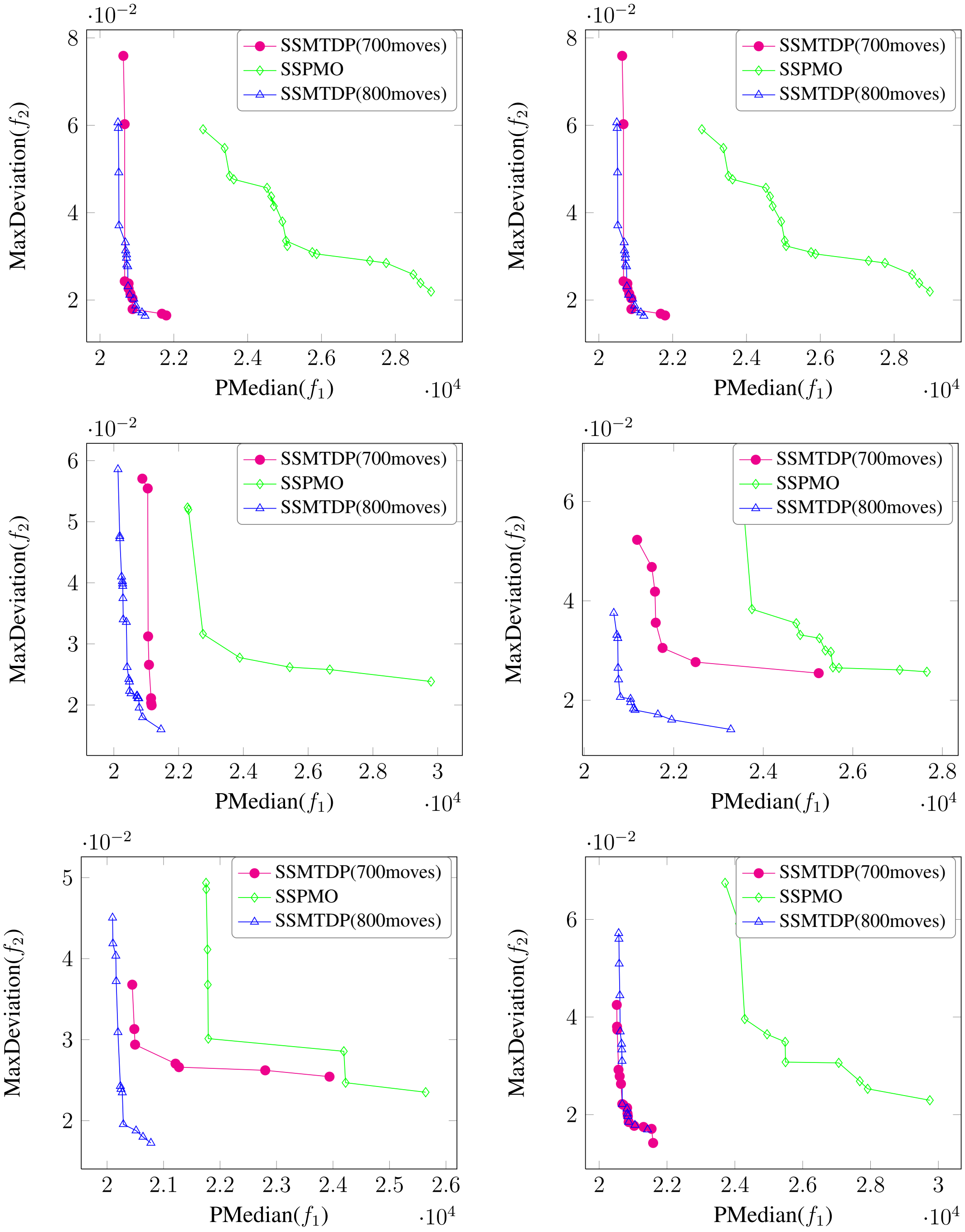

Figure 7.10: Comparison of Pareto fronts, SSPMO vs. SSMTDP. Instances from (500,20), part 1.

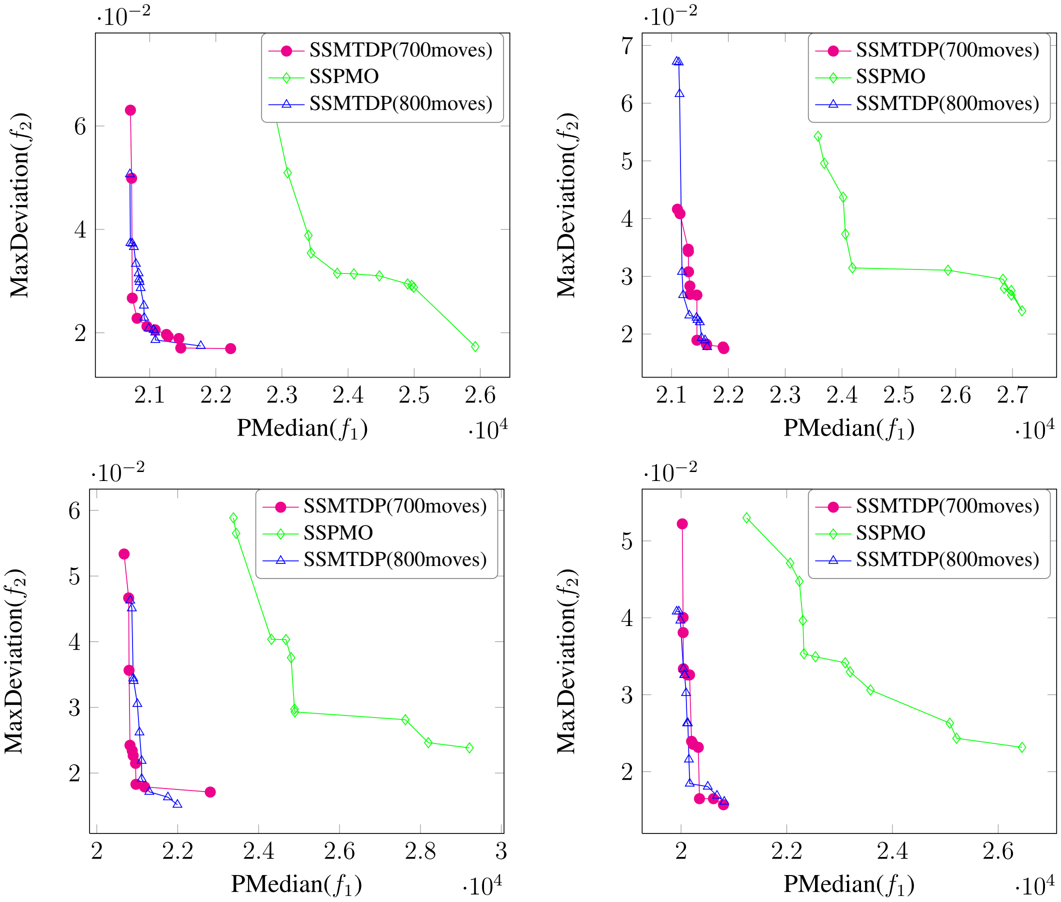

Figure 7.11: Comparison of Pareto fronts, SSPMO vs. SSMTDP. Instances from (500,20), part 2.

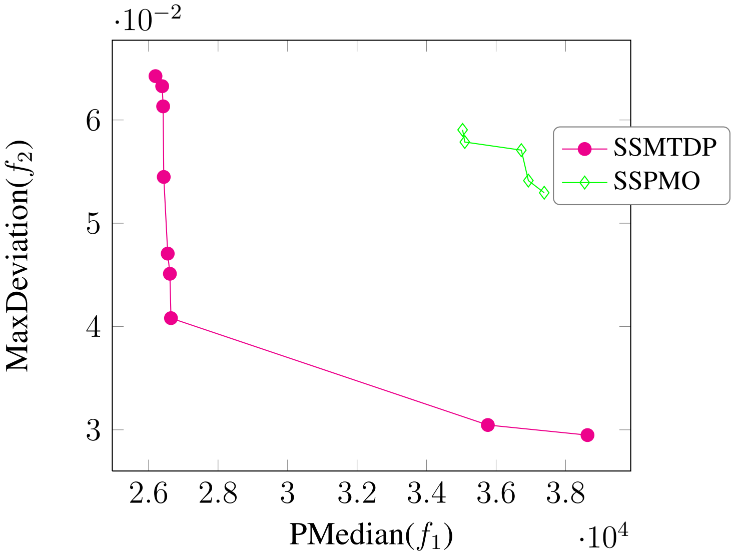

Figure 7.12: Comparison of Pareto fronts, SSPMO vs. SSMTDP. Instance with 1000 BUs and 50 territories.

efficient solutions in less than 3 hours, note that all of them are feasible solutions. Therefore, the proposed heuristic procedures are a very attractive alternative for the firm since these procedures are able to generate more than one attractive solution for the decision maker.

7.6 Conclusions

In this Chapter, a computational evaluation of the proposed heuristic procedures was carried out. This evaluation was based on well-known performance measures used in multiobjective optimization, such as number of points, size of the space covered (SSC), k-distance, and coverage of two sets measure. The procedures were applied to two different instance sets of $(n, p) \in \{$(500,20),(1000,50)$\}$. For each of these sets, 10 instances were randomly generated based on real-world data provided by the industrial partner.

During the evaluation of the GRASP strategies, an ANOVA was carried out for each performance measure: number of points, k-distance (mean), k-distance (max), and SSC. It was observed that only the SSC measure presents significant variation. The worst behavior was for TGRASP-II. In contrast, the number of points and k-distance did not have significant changes independently of the used GRASP procedure.

According to the coverage of two sets measure, the best GRASP strategy is BGRASP-II, this

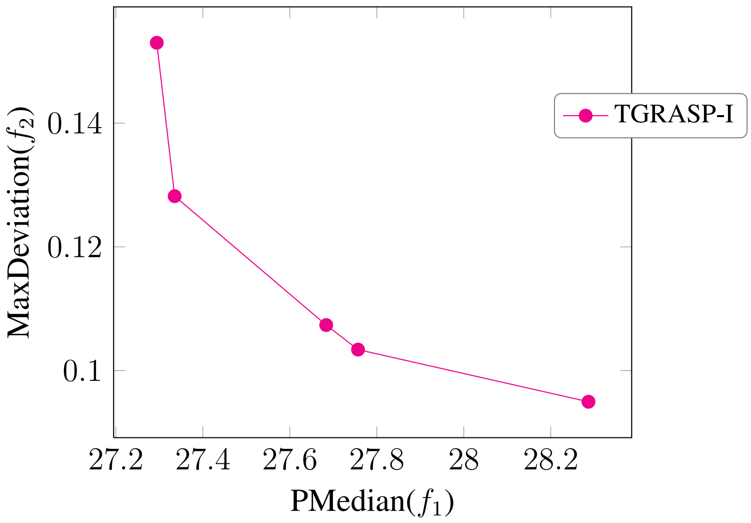

Figure 7.13: Pareto front for a real-world case with 1999 BUs and 50 territories, TGRASP-I

procedure dominates the highest proportion of efficient points given by BGRASP-I, TGRASP-I and TGRASP-II. In contrast, when the time is the most important performance measure, BGRASP-I showed the best behavior.

All GRASP strategies were compared with an evolutionary algorithm (NSGA-II) widely used in multiobjective programming. The proposed GRASP strategies outperform the NSGA-II procedure over all instances tested. This reveals the fact that heuristics specifically developed to exploiting the problem structure have better results than those generic procedures, particularly in problems with very difficult constraints such as connectivity constraints in territory design.

BGRASP-I was integrated as a diversification method into a SS scheme developed in this dissertation (SSMTDP). The SSMTDP procedure showed excellent performance when it was compared with the SSPMO algorithm.

In addition, approximated efficient solutions were obtained by applying TGRASP-I to a real-world case with 1999 BUs and 50 territories. This case was previously solved by the firm; however, they reported an infeasible solution with respect to the balancing requirement of the sales volume as the best solution. Therefore, the proposed procedures are a good alternative for the industrial partner given that these are capable to generate feasible solutions in a short time. In summary, empirical work reveals that these procedures are capable to find not only one but a set of good solutions that provide alternatives of territory designs to the decision maker. The obtained results for the real case are better than the single result generated by the firm.

CHAPTER 8

CONCLUSIONS

In this dissertation a commercial territory design problem is addressed. During the literature review some opportunity areas were detected. For instance, it was observed that there was not an exact solution procedure for solving the single-objective version of this problem. To the best of my knowledge, there was not a quadratic formulation for any other application of territory design. An exact solution procedure that allows to obtain global optimal solutions for MILP formulations and local optimal solutions for the proposed IQP formulation of this problem was successfully applied.

The multiobjective territory design problem was modeled as a bi-objective optimization model where dispersion and maximum deviation with respect to the average of the number of customers are considered as objective functions. These objectives are subject to multiple constraints such as exclusive assignment, creation of a fixed number of territories, balancing of sales volume, and connectivity. In this dissertation, the proof of NP-Completeness for the problem addressed was developed. Exact and heuristic procedures were proposed for solving the bi-objective commercial territory design problem. The exact solution procedure is an integration of a cut generation procedure and the ε-constraint method. Two variants of the ε-constraint were implemented in this dissertation.

Empirical evaluation of the proposed exact and heuristic procedures was carried out. The proposed heuristic procedures outperformed two of the well-known and most respected multiobjective algorithms (NSGA-II and SSPMO). In addition,a real-world case from a beverage distribution company was solved, obtaining significantly better results than those obtained by the firm. A summary of the research contributions is in the following section. This dissertation work finishes with some lines for future research.

8.1 Summary of Research Contributions

Although several territory design approaches have appeared in the literature, the specific features present in this concrete problem make it very unique, and not addressed before to the best of my knowledge. In general, the main contributions of this work are the following:

- Design and development of metaheuristic algorithms for successfully addressing large-scale instances of this problem. This includes efficient exploitation of problem structure and development of algorithmic components and intelligent search strategies.
  - The introduction of four GRASP strategies (called BGRASP-I, BGRASP-II, TGRASP-I, and TGRASP-II) for solving the bi-objective version of the problem.
  - A novel heuristic procedure called SSMTDP which is based on Scatter Search metaheuristic, whose components have been intelligently designed.
  - A combination method that consists of a hybrid approach. It provides quality and diversification of solutions. This procedure can be used in other applications such as clustering.
- Introduction of the first bi-objective commercial territory design model.
- Derivation of valid inequalities for strengthening the problem formulations.
- Proof of NP-Completeness for two variants of the commercial territory design problem.
- Design and development of an efficient exact method for obtaining efficient Pareto fronts for medium-size instances of the problem. The procedure is based on an integration of the well-known ε-constraint method and a cut generation strategy, that initially drops the (exponential number of) connectivity constraints.
- As a by-product, this dissertation also makes a contribution to the single-objective version of the problem such as the development of a new quadratic model that uses far fewer binary variables and it is a lot faster to solve by mixed integer non-linear solvers. A second contribution to the single-objective version is the development of an exact optimization scheme. Prior to this, no exact optimization method had been developed for single-objective versions of the problem.

- An additional alternative for solving the single-objective version through a hybrid algorithm that uses the quadratic formulation of this problem. The quadratic model has proved in the experimental work that instances of size around 400 or 500 basic units and 20 territories can be solved relatively quickly. So, a heuristic strategy that partitions the original problem into smaller problems and then integrates the individual solutions to these problems is attractive.

- In the literature of territory design, it is one of the few works dealing with a bi-objective problem subject to connectivity constraints.

- Most of the works in multiobjective optimization use a weighted sum function that aggregates the multiple-criteria in a single function. Then, the multiobjective problem is treated as a single-objective problem. In this work, the multiobjective problem is genuinely studied as such.

- The proposed heuristic procedures were evaluated by using a real-world instance provided by the industrial partner. It was observed that the procedures are capable to find efficient solutions in a short time. These solutions were better than the one obtained by the firm. Thus, the proposed solution procedures are a good alternative for providing more than one alternative solution to the decision maker.

8.2 Directions for Future Research

The proposed heuristic procedures have been proved in the bi-objective version of the commercial territory design problem. However, the problem can be addressed by optimizing one more objective such as the balancing with respect to the sales volume. In the other hand, due that the territory design problem takes place in a previous stage of the routing of product, the routing cost is another requirement that can be incorporated to the current models. This requirement could be treated as objective or as a constraint. The current procedures can be used as a basis to develop new solution techniques for solving the new problems.

The problem addressed in this dissertation was studied from a deterministic point of view. However, there are some parameters such as the sales volume that have stochastic nature. Therefore, another extension of this problem can be done by considering stochastic demand. So, the new problem should be addressed by using another solution methods that belong to the stochastic optimization field.

The efficiency of the solution procedures have been proved, however these can be improved by modifying some of their components:

- In the exact optimization method, the development of a branch-and-cut procedure that exploit the problem structure is open. The solution technique proposed in this work solves the problem, but it is not efficient since many complete branch-and-bound iterations are needed. Within a branch-and-cut framework, a single iteration of the branch-and-bound method is executed. The effort is then in generating valid facet-defining inequalities from the polyhedral theory perspective.

- During the development of GRASP strategies, an adaptive memory strategy was implemented. During the construction phase, specifically for selecting the seed nodes, a counter (r) for computing the number of iterations in which a node is chosen as a seed for opening a territory was implemented. Then, in the following iterations the probability for selecting a node as initial seed is computed by taking into account the value of r. Thus, large values of r has less probability of being selected. During the experimental work, the same results were reported by GRASP with or without this particular adaptive memory strategy. However, the exploration of other adaptive memory strategies may yield better results.

- Another feature that can be incorporated to the current GRASP strategies is that of reactivity which has reported good results in single-objective models.

- The development of other neighborhoods that would allow a different explorations of the search space may also proven worthwhile. For instance, interchanging of nodes between two adjacent territories and/or developing an iterated local search such that a subset of the territories in a current solution is re-aligned. These new neighborhoods will allow to provide more diversity during the searching process. In the current process only one type of neighborhood was developed.

- In the current combination method it is possible to change the objective function of the assignment subproblem which is solved for matching the territory centers. An alternative can be the bottleneck matching problem objective.

- There may be other ideas for combining solutions. For instance, path relinking could be a good strategy for this task.

Finally, the proposed heuristic procedures can be applied to other kind of territory design applications such as political districting. Evidently, some components need to be adapted to the specific application.

Bibliography

[1] J. E. Souza de Cursi A. G. N. Novaes, A. C. L. da Silva, and J. C. Souza. Solving continuous location-districting problems with voronoi diagrams. *Computers & Operations Research*, 36(1):40–59, 2009.

[2] F. Bação, V. Lobo, and M. Painho. Applying genetic algorithms to zone design. *Soft Computing*, 9(5):341–348, 2005.

[3] R. P. Beausoleil. MOSS multiobjective scatter search applied to non-linear multiple criteria optimization. *European Journal of Operational Research*, 169(2):426–449, 2006.

[4] P. Bertolazzi, L. Bianco, and S. Ricciardelli. A method for determining the optimal districting in urban emergency services. *Computers & Operations Research*, 4(1):1–12, 1977.

[5] M. Blais, S. D. Lapierre, and G. Laporte. Solving a home-care districting problem in an urban setting. *Journal of the Operational Research Society*, 54(11):1141–1147, 2003.

[6] C. W. Bong and Y. C. Wang. A multiobjective hybrid metaheuristic approach for GIS-based spatial zone model. *Journal of Mathematical Modelling and Algorithms*, 3(3):245–261, 2004.

[7] R. Bowerman, B. Hall, and P. Calamai. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. *Transportation Research Part A*, 29(2):107–123, 1995.

[8] B. Bozkaya, E. Erkut, and G. Laporte. A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1):12–26, 2003.

[9] R. Caballero, X. Gandibleux, and J. Molina. MOAMP: A generic multiobjective metaheuristic using an adaptive memory. Technical report, University of Valenciennes, Valenciennes, France, 2004.

[10] S. I. Caballero-Hernández, R. Z. Ríos-Mercado, F. López, and E. Schaeffer. Empirical evaluation of a metaheuristic for commercial territory design with joint assignment constraints. In J. E. Fernandez, S. Noriega, A. Mital, S. E. Butt, and T. K. Fredericks, editors, *Proceedings of the 12th Annual International Conference on Industrial Engineering Theory, Applications, and Practice (IJIE)*, pages 422–427, Cancun, Mexico, November 2007. ISBN: 978-0-9654506-3-8.

[11] F. Caro, T. Shirabe, M. Guignard, and A. Weintraub. School redistricting: Embedding GIS tools with integer programming. *Journal of the Operational Research Society*, 55(8):836–849, 2004.

[12] A. Chinchuluun and P. M. Pardalos. A survey of recent developments in multiobjective optimization. *Annals of Operations Research*, 152(1):29–50, 2007.

[13] C.-I. Chou, Y.-L. Chu, and S.-P. Li. Evolutionary strategy for political districting problem using genetic algorithm. In Y. Shi, G. D. van Albada, and J. Dongarra, editors, *Computational Science – ICCS 2007*, volume 4490 of *Lecture Notes in Computer Science*, pages 1163–1166. Springer, Berlin, Germany, 2007.

[14] C.-I. Chou and S.-P. Li. Spin systems and political districting problem. *Journal of Magnetism and Magnetic Materials*, 310(2):2889–2891, 2007.

[15] Y. Collete and P. Siarry. *Multiobjective Optimization: Principles and Case Studies*. Springer, Berlin, Germany, 2004.

[16] S. J. D'Amico, S.-J. Wang, R. Batta, and C. M. Rump. A simulated annealing approach to police district design. *Computers & Operations Research*, 29(6):667–684, 2002.

[17] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elistist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, booktitle =.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyerivan. A fast elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[19] E. Domínguez and J. Mu noz. A neural model for the p-median problem. *Computers & Operations Research*, 35(2):404–416, 2008.

[20] A. Drexl and K. Haase. Fast approximation methods for sales force deployment. *Management Science*, 45(10):1307–1323, 1999.

[21] E. I. Ducheyne, B. De Baets, and R. R. De Wulf. Fitness inheritance in multiple objective evolutionary algorithms: A test bench and real-world evaluation. *Applied Soft Computing*, 8(1):337–349, 2008.

[22] M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, Germany, 2005.

[23] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, 22(4):425–460, 2000.

[24] M. Ehrgott and X. Gandibleux. Approximative solution methods for multiobjective combinatorial optimization. *Top*, 12(1):1–90, 2004.

[25] M. Ehrgott and S. Ruzika. Improved ε-constraint method for multiobjective programming. *Journal of Optimization Theory and Applications*, 138(3):375–396, 2008.

[26] E. Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60, 1990.

[27] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.

[28] J. A. Ferlang and G. Gunette. Decision support system for the school districting problem. *Operations Research*, 38(1):15–21, 1990.

[29] B. Fleischmann and J. N. Paraschis. Solving a large scale districting problem: A case report. *Computers & Operations Research*, 15(6):521–533, 1988.

[30] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freedman and Company, New York, USA, 2000.

[31] R. S. Garfinkel and G. L. Nemhauser. Solving optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):B495–B508, 1970.

[32] F. Glover. Heuristics for integer programming usign surrogate constraints. *Decision Sciences*, 8(1):156–166, 1977.

[33] C. Gomes, J. Climaco, and J. R. Figueira. A scatter search method for the bi-criteria multi-dimensional $\{0,1\}$-Knapsack problem using surrogate relaxation. *Journal of Mathematical Modelling and Algorithms*, 3(3):183–208, 2004.

[34] J. Guo, G. Trinidad, and N. Smith. MOZART: A multi-objective zoning and aggregation tool. In *Proceedings of the Philippine Computing Science Congress (PCSC)*, pages 197–201, Manila, Philippines, 2000.

[35] Y. Haimes, L. Lasdon, and D. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1(3):296–297, 1971.

[36] S. W. Hess and S. A. Samuels. Experiences with a sales districting model: Criteria and implementation. *Management Science*, 18(4):998–1006, 1971.

[37] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau. Nonpartisan political redistring by computer. *Operations Research*, 13(6):998–1006, 1965.

[38] M. Hojati. Optimal political districting. *Computers & Operations Research*, 23(12):1147–1161, 1996.

[39] D. L. Horn, C. R. Hampton, and A. J. Vandenberg. Practical application of district compactness. *Political Geography*, 12(2):103–120, 1993.

[40] J. Horn, N. Nafploitis, and D. E. Goldberg. A niched Pareto genetic algorithm for multi-objective optimization. In Z. Michalewicz, editor, *Proceedings of the First IEEE Conference on Evolutionary Computation*, volume 1, pages 82–87, Piscataway, USA, 1994. IEEE Service Center.

[41] ILOG, Inc., Mountain View, USA. *ILOG CPLEX 11.0 User's Manual*, 2007.

[42] D. S. Johnson. The NP-completeness column: An ongoing guide. *Journal of Algorithms*, 3:185–192, 1982.

[43] J. Kalcsics, S. Nickel, and M. Schröder. Towards a unified territorial design approach: Applications, algorithms, and GIS integration. *Top*, 13(1):1–56, 2005.

[44] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. I: the p-centers. *SIAM Journal on Applied Mathematics*, 37(3):513–538, 1979.

[45] J. Knowles and D. Corne. The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimisation. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 98–105, Piscataway, USA, 1999. IEEE Service Center.

[46] G. R. Kocis and I. E. Grossmann. Relaxation strategy for the structural optimization of process flow sheets. *Industrial & Engineering Chemistry Research*, 26(9):1869–1880, 1987.

[47] G. R. Kocis and I. E. Grossmann. Computational experience with DICOPT solving MINLP problems in process systems engineering. *Computers and Chemical Engineering*, 13(3):307–315, 1989.

[48] A. Konak, D. W. Coit, and A. E. Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.

[49] M. Laguna and R. Martí. *Scatter Search: Methodology and Implementations in C*. Kluwer, Boston, USA, 2003.

[50] P.G. Marlin. Application of the transportation model to a large-scale districting problem. *Computers & Operations Research*, 8(2):83–96, 1981.

[51] R. Martí, M. Laguna, and F. Glover. Principles of scatter search. *European Journal of Operational Research*, 169(2):359–372, 2006.

[52] A. Mehrotra, E. L. Johnson, and G. L. Nemhauser. An optimization based heuristic for political districting. *Management Science*, 44(8):1100–1113, 1998.

[53] R. Minciardi, P. P. Puliafito, and R. Zoppoli. A districting procedure for social organizations. *European Journal of Operational Research*, 8(1):47–57, 1981.

[54] J. Molina, R. Martí, and R. Caballero. SSPMO: A scatter tabu search procedure for non-linear multiobjective optimization. *INFORMS Journal on Computing*, 19(1):91–100, 2007.

[55] L. Muyldermans, D. Cattryse, D. Van Oudheusden, and T. Lotan. Districting for salt spreading operations. *European Journal of Operational Research*, 139(3):521–532, 2002.

[56] R. G. Niemi, B. Grofman, C. Carlucci, and T. Hofeller. Measuring compactness and the role of a compactness standard in a test for partisan and racial gerrymandering. *Journal of Politics*, 52(4):1155–1181, 1990.

[57] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, New York, USA, 1998.

[58] F. Pezzella and B. Nicolleti. A system approach to the optimal health-care districting. *European Journal of Operational Research*, 8(2):139–146, 1981.

[59] A. R. Rahimi-Vahed, M. Rabbani, R. Tavakkoli-Moghaddam, S. A. Torabi, and F. Jolai. A multi-objective scatter search for mixed-model assembly line sequencing problem. *Advanced Engineering Informatics*, 21(1):85–99, 2007.

[60] M. G. C. Resende and R. F. Werneck. A hybrid heuristic for the p-median problem. *Journal of Heuristics*, 10(1):59–88, 2004.

[61] F. Ricca. A multicriteria districting heuristic for the aggregation of zones and its use in computing origin-destination matrices. *Information Systems and Operational Research*, 42(1):61–77, 2004.

[62] F. Ricca and B. Simeone. Local search algorithms for political districting. *European Journal of Operational Research*, 189(3):1409–1426, 2008.

[63] R. Z. Ríos-Mercado and E. A. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36(3):755–776, 2009.

[64] L. V. Santana-Quintero, N. Ramírez, and C. Coello. A multi-objective particle swarm optimizer hybridized with scatter search. In *MICAI 2006: Advances in Artificial Intelligence*, volume 4293 of *Lecture Notes in Computers Science*, pages 294–304. Springer, Berlin,Germany, 2006.

[65] M. N. Scott, R. G. Cromley, and E. K. Cromley. Multi-objective analysis of school district regionalization alternatives in Connecticut. *The Professional Geographer*, 48:1–14, 1996.

[66] M. Segal and D. B. Weinberger. Turfing. *Operations Research*, 25(3):367–386, 1977.

[67] J. A. Segura-Ramiro, R. Z. Ríos-Mercado, A. M. Álvarez-Socarrás, and K. de Alba Romenus. A location-allocation heuristic for a territory design problem in a beverage distribution firm.

In J. E. Fernandez, S. Noriega, A. Mital, S. E. Butt, and T. K. Fredericks, editors, *Proceedings of the 12th Annual International Conference on Industrial Engineering Theory, Applications, and Practice (IJIE)*, pages 428–434, Cancun, Mexico, November 2007. ISBN: 978-0-9654506-3-8.

[68] T. Shirabe. Classification of spatial properties for spatial allocation modeling. *Geoinformatica*, 9(3):269–287, 2005.

[69] T. Shirabe. A model of contiguity for spatial unit allocation. *Geographical Analysis*, 37(1):2–16, 2005.

[70] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, UK, 1986.

[71] F. Tavares-Pereira, J. R. Figueira, V. Mousseau, and R. Bernard. Multiple criteria districting problems. The public transportation network pricing system of the Paris region. *Annals of Operations Research*, 154(1):69–92, 2007.

[72] F. Tavares-Pereira, J. R. Figueira, V. Mousseau, and R. Bernard. Comparing two territory partitions in districting problems: Indices and practical issues. *Socio-Economic Planning Sciences*, 43(1):72–88, 2009.

[73] J. Viswanathan and I. E. Grossmann. A combined penalty function and outer approximation method for MINLP optimization. *Computers and Chemical Engineering*, 14(7):769–782, 1990.

[74] T. Westerlund and F. Pettersson. An extended cutting plane method for solving Convex MINLP problems. *Computers and Chemical Engineering*, 19(1):131–136, 1995.

[75] T. Westerlund and R. Pörn. Solving pseudo-convex mixed integer optimization problems by cutting plane techniques. *Optimization and Engineering*, 3(3):253–280, 2002.

[76] T. Westerlund, H. Skrifvars, I. Harjunkoski, and R. Pörn. An extended cutting plane method for solving a class of non-convex MINLP problems. *Computers and Chemical Engineering*, 22(3):357–365, 1998.

[77] E. Ziztler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[78] E. Ziztler, M. Laumanns, and L. Thiele. SPEA2: Improving the strenght Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.

[79] E. Ziztler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[80] A. A. Zoltners and P. Sinha. Sales territory alignment: A review and model. *Management Science*, 29(11):1237–1256, 1983.

[81] A. A. Zoltners and P. Sinha. Sales territory design: Thirty years of modeling and implementation. *Marketing Science*, 24(3):313–331, 2005.

APPENDIX A

MULTIOBJECTIVE OPTIMIZATION

Multiobjective optimization allows a degree of freedom not present in single-objective optimization. In multiobjective optimization the usual meaning of optimum makes no sense because a solution optimizing all objectives simultaneously does not exist in general. Instead, a search for a feasible solution yielding the best compromise among objectives on a set of solutions is performed. These solutions are called *efficient* solutions, and the set of images of these solutions is called the *tradeoff surface* or *Pareto optimal*. The multiple objectives add to the difficulty of combinatorial optimization problems makes these problems are very hard to solve exactly [23], even if they are derived from easy single-objective optimization problems.

A.1 BASIC CONCEPTS AND NOTATION

A general multiobjective optimization problem is defined as follows

$$\begin{array}{rl} \text{(MOP)} \quad \min & (f_1(x), ..., f_p(x)) \\ \text{subject to:} & x \in X, \end{array}$$

where $X \subset \mathbb{R}^n$ is a feasible set and $f : \mathbb{R}^n \to \mathbb{R}^p$ is a vector valued objective function. By $Y = f(X) \subset \mathbb{R}^p$ we denote the image of the feasible set in the objective space. Efficiency refers to solutions x in the decision space. In terms of the objective space, with objective vectors $f(x) \in \mathbb{R}^p$ we use the notion of non-dominance: If x is efficient then $f(x) = (f_1(x), ..., f_p(x))$ is called non-dominated (or also efficient). The following notation (see [25]) is used to define ordering relations on $\mathbb{R}^p$. For $f(x), f(x') \in \mathbb{R}^p$,

- $f(x) < f(x')$ denotes $f_k(x) < f_k(x')$ for all $k = 1, ..., p$,
- $f(x) \leqq f(x')$ denotes $f_k(x) \leq f_k(x')$ for all $k = 1, ..., p$,

- $f(x) \leq f(x')$ denotes $f_k(x) \leqq f_k(x')$ but $f(x) \neq f(x')$.

The concept of Pareto optimality or efficiency is based on these ordering relations. Thus, a point $x \in X$ is called:

1. a *weakly efficient* solution if there is no $x' \in X$ such that $f(x') < f(x)$,
2. an *efficient* solution if there is no $x' \in X$ such that $f(x') \leq f(x)$,
3. a *strictly efficient* solution if there is no $x' \in X$, $x \neq x'$, such that $f(x') \leqq f(x)$.

We denote the sets of weakly efficient solutions, efficient solutions, and strictly efficient solutions by X_{wE}, X_{E}, and X_{sE}, respectively. The images for X_{wE} and X_{E} are called weakly non-dominated points (Y_{w}) and non-dominated points (Y_{E}), respectively. Note that, strictly efficient solutions correspond to unique efficient solutions, i.e.,

$$x \in X_{\text{sE}} \Leftrightarrow x \in X_{\text{E}}$$

and

$$\big|\{x' : f(x') = f(x)\}\big| = 1.$$

From the definitions we obtain the following relations $Y_{\text{N}} \subset Y_{\text{wN}}$ and $X_{\text{sE}} \subset X_{\text{E}} \subset X_{\text{wE}}$ (see [22]). In particular, when the multiobjective problem has integer variables and linear set of constraints and objective functions, we can define a multiobjective combinatorial optimization (MOCO) problem as follows:

$$\begin{array}{lll} \text{(MOCO)} & \min & (Cx) \\ & \text{subject to:} & Ax \geq b, x \in \mathbb{Z}^n, \end{array}$$

where C is a $p \times n$ objective function matrix, and c^k denotes the k-th row of C. A is an $m \times n$ matrix of constraint coefficients and $b \in \mathbb{R}^m$. Usually, the entries of C, A and b are integers. The feasible set $X = \{Ax \geq b, x \in \mathbb{Z}^n\}$ may describe a combinatorial structure such as paths and matchings. We shall assume that X is a finite set. By $Y = CX$ we denote the image of X under C in $\mathbb{R}^p$.

A.2 Exact Methods

The traditional approach for solving MOPs is by scalarization, which involves formulating a multiobjective problem as a single-objective problem by means of a real-valued scalarizing function typically being a function of the objective functions of the MOP, auxiliary scalar or vector variables, and/or scalar or vector parameters. Sometimes the feasible set of the MOP is additionally restricted by new constraint functions related to the objective functions of the MOP and/or the new variables introduced. For more details on scalarizing (and non-scalarizing) techniques, see [22] and [15].

A.2.1 The Weighted Sum Method

The biggest additional challenge in solving MOCOs as compared to multiobjective linear programs (MOLPs)

$$\min\{Cx : Ax \geq b, x \geqq 0\}$$

results from the existence of efficient solutions which are not optimal for any scalarization technique using weighted sums (A.1).

$$\min_{x \in X} \sum_{k=1}^{p} \lambda_k f_k(x) \tag{A.1}$$

These solutions are called *unsupported efficient* solutions X_{NE}. Those that are optimal for some weighted sum problem are called *supported efficient* solutions X_{SE}.

Note that, in the context of heuristics, an exact algorithm which finds all supported solutions, i.e., solves (A.1) for all $\lambda \in \Lambda = \{\lambda \in \mathbb{R}^p_> : \sum_{k=1}^{p} \lambda_j = 1\}$, becomes a heuristic for determination of X_{E}. So, unsupported efficient solutions are the main reason why the computation of X_{E} becomes too hard.

The weighted sum method is the simplest method for solving multiobjective optimization problems. However, for nonconvex problems it may work poorly given that it is not able to obtain the unsupported efficient solutions.

A.2.2 THE ε-CONSTRAINT METHOD

Another scalarization technique which can be applied when X and Y are not convex is the well-known ε-constraint method. In this method there is no aggregation criteria, instead only one of the original objectives is minimized, while the others are transformed to constraints. That is,

$$
\begin{array}{lrl}
(\varepsilon\text{-C}) & \displaystyle\min_{x\in X} & f_k(x) \\
& \text{subject to:} & f_j(x) \leq \varepsilon_j \quad j \neq k,
\end{array}
$$

where $\varepsilon = (\varepsilon_1, ..., \varepsilon_{k-1}, \varepsilon_{k+1}, ..., \varepsilon_p)^T \in \mathbb{R}^{p-1}$ and $k \in \{1, ..., p\}$. A feasible solution of (ε-C) problem is denoted by $X_k^\varepsilon = \{x \in X : f_i(x) \leq \varepsilon_i, i \neq k\}$.

Optimal solutions of (ε-C) are *weakly efficient*. These can be characterized as efficient or strictly efficient; however, the characterization process is not an easy process.

Optimal solutions of (ε-C) are weakly efficient solutions of the MOP, and it has the weakness that there is not an easy way to check the conditions for the characterization of these solutions as efficient solutions. Recently, Erhgott and Ruzika [25] present a modification of (ε-C) by including nonnegative slack variables in the traditional formulation (ε-C). The resulting model is given by (ε^+-C) model,

$$
\begin{array}{lrl}
(\varepsilon^{+}\text{-C}) & \displaystyle\min_{x\in X} & f_k(x) - \sum_{i\neq k} \lambda_i s_i \\
& \text{subject to:} & f_j(x) + s_j \leq \varepsilon_j \\
& & s_j \geq 0, j \neq k,
\end{array}
$$

where $\varepsilon = (\varepsilon_1, ..., \varepsilon_{k-1}, \varepsilon_{k+1}, ..., \varepsilon_p)^T \in \mathbb{R}^{p-1}$, $k \in \{1, ..., p\}$ and $\lambda_j \geq 0, j \neq k$ are nonnegative weights.

Erhgott and Ruzika [25] present the proof of the following theorem

Theorem A.1. Let (x', s') be an optimal solution of (ε^+-C) with $\lambda > 0$. Then, x' is an *efficient* solution of the MOP.

This result gave us the motivation to implement the modified ε-C method for obtaining efficient solutions to the bi-objective commercial territory design problem addressed in this work.

A.3 Metaheuristics

Approximation methods are an attractive alternative for solving complex MOPs where the exact methods have poor performance. In multiobjective optimization an approximation method finds either sets of locally potentially efficient solutions, that are merged to form a set of potentially efficient solutions - the approximation- or globally efficient solutions according to the current approximation [24].

A metaheuristic can be defined as a powerful technique that can be adapted to solve a large number of problems. Thus, a metaheuristic refers to an iterative master strategy that combines different components for exploring the search space. There is a large family of metaheuristics, for instance, Greedy Randomized Adaptive Search Procedure (GRASP), Scatter Search (SS), Tabu Search (TS), Genetic Algorithm (GA), and Neural Networks (NN). Erghgott and Gandibleux [24] present an extensive survey of approximation methods for MOCO problems.

A.4 Performance Measures

There are different performance measures used to evaluate the quality of those approximated efficient solutions obtained by approximation procedures in multiobjective optimization. In the literature of multiobjective optimization, the most used performance measures are the following:

1. *Number of points:* It is an important measure because efficient frontiers that provide more alternatives to the decision maker are preferred than those frontiers with few efficient points.

2. *k-distance:* This density-estimation technique used by Zitzler, Laumanns, and Thiele [78] in connection with the computational testing of SPEA2 is based on the kth-nearest neighbor method of Silverman [70]. This metric is simply the distance to the kth-nearest efficient point. So, the smaller the k-distance the better in terms of the frontier density.

3. *Size of space covered (SSC):* This metric was suggested by Zitzler and Thiele [79]. This measure computes the volume of the dominated points. Hence, the larger the value of SSC the better.

4. *C(A,B):* It is known as the coverage of two sets measure [79]. This measure represents the proportion of points in the estimated efficient *B* that are dominated by the efficient points in the estimated frontier *A*.