GRASP Strategies for a Bi-objective Commercial Territory Design Problem

M. Angélica Salazar-Aguilar
Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León, Mexico
e.mail: angy@yalma.fime.uanl.mx

Roger Z. Ríos-Mercado
Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León, Mexico
e.mail: roger@yalma.fime.uanl.mx

José Luis González-Velarde
Center for Quality and Manufacturing
Tecnológico de Monterrey, Mexico
e.mail: gonzalez.velarde@itesm.mx

January 2010

Abstract

In this work a problem motivated by a real-world case from a beverage distribution firm in Mexico is addressed. Different planning criteria are taken into account in order to create acceptable territory designs. Namely, each territory needs to be compact, connected and balanced according to two attributes (number of costumers and product demand). Two GRASP based heuristics (BGRASP and TGRASP) are proposed for this NP-hard combinatorial optimization problem. For each of them two variants are studied: i) keeping connectivity as a hard constraint during construction and post-processing phases and, ii) ignoring connectivity during the construction phase and adding this as a minimizing objective function during the post-processing phase . The main difference between BGRASP and TGRASP is the way they consider the planning criteria during the construction phase. In BGRASP, the construction attempts to find high quality solutions based on the optimization of two criteria: compactness and balancing according to the number of customers, that is, demand is treated as a constraint. The construction phase in TGRASP considers three objectives to be optimized: compactness and balancing with respect to the two attributes (number of customers and demand). That is, the demand balancing constraints are relaxed and treated as part of the objective function. The proposed procedures are evaluated on a variety of problem instances, with 500 and 1000 basic units. We carried out an analysis of these procedures using different performance measures such as number of non-dominated points, k-distance, size of the space cover (SSC), coverage of two sets measure, and time. We observed that, SSC, coverage of two sets measure and time, present significant variation depending on the GRASP procedure used. The number of points and k-distance measures did not present significant variation over all evaluated procedures. BGRASP-I provides good frontiers in shortest time and BGRASP-II has the best coverage of the efficient points given by the others procedures.

Keywords: combinatorial optimization, bi-objective programming, territory design, GRASP.

1 Introduction

The problem addressed in this paper arises from a beverage distribution firm in Mexico. Single objective versions of this problem have been studied by Ríos-Mercado and Fernández [9] and Segura-Ramiro et al. [11]. The introduction of new bi-objective models and an exact solution procedure to this problem was proposed recently [10]. In general, commercial territory design belongs to the family of districting problems that have a broad range of applications like political districting [1], school districting [5, 3], and sales and service territory design [6, 16]. In most of these applications a single objective problem is considered, however in the real world it is very common to pursue more than one criterion. In fact, looking at the literature on territory design (TD), few works address these problems as multi-objective problems [13, 8, 10]. Territory design (TD) is very common in every enterprise dedicated to product sales and product distribution, specifically when the firm needs to divide the market into smallest regions to delegate responsibilities to facilitate the sales and distribution of goods. These decisions need to be constantly evaluated due to the frequent market changes such as the introduction of new products or changes in the workload, which are factors that affect the territory design. Additionally, the multiple planning requirements that the firm wants to satisfy and the large amount of customers that need to be grouped makes this difficult task even more critical. An efficient tool with capacity to provide good solutions to large problems is needed. In [10], we proposed an exact solution procedure for the problem addressed in this work. They reported efficient solutions for instances with up to 150 BUs and 6 territories were reported. In the real world, it is very common to deal with instances of 500 to 2000 BUs. This motivates the heuristics proposed in our work. In this work we propose some heuristic procedures based on GRASP strategies (BGRASP and TGRASP) aiming at finding a good approximation of the Pareto frontier. Each of these strategies consists of two main phases: construction and post-processing. In the construction phase a simultaneous territory creation is carried out and in the post-processing phase the neighborhood is explored in a similar way to that of the MOAMP procedure applied by Molina, Martí, and Caballero [7]. We tested the proposed procedures on a set of instances based on real-world cases. The tests indicate that BGRASP has better performance than TGRASP. The paper is organized as follows. In Section 2 the problem is described and a bi-objective optimization model is presented. Section 3 shows details about the proposed solution procedures and Section 4 includes the experimental work. Finally we wrap up with the conclusions in Section 5.

2 Multi-objective Commercial Territory Design

2.1 Problem Description

In particular, the problem consists of finding a partition of the entire set of city blocks or basic units (BUs) into a fixed number (p) of territories, considering several planning territory requirements such as compactness, balance and connectivity. Compactness means customers within a territory should be relatively close to each other. Balance implies territories with similar size with respect to two attributes (number of customers and sales volume). Connectivity means BUs in the same territory can reach each other without leaving the territory. In addition, exclusive assignment from BUs to territories is needed. The problem is modeled by an undirected graph $G = (V, E)$, where V is the set of nodes (BUs) and E is the set of edges representing adjacency between blocks (BUs). That is, a block or BU j is associated with a node, and an edge connecting nodes i and j exists if i and j are adjacent. For each node $j \in V$ there are some associated parameters such as geographical coordinates (c_x, c_y), and two measurable attributes (number of customers and sales volume) are defined. The number of territories is given by parameter p. It is required that each node is assigned to only one territory (exclusive assignment). The company wants balanced territories with respect to each of the attribute measures. Let us define the size of territory V_k with respect to attribute a as: $w^{(a)}(V_k) = \sum_{i \in V_k}(w_i^{(a)})$, where $a \in \{1, 2\}$ and $w_i^{(a)}$ is the value associated to attribute a in node $i \in V$. Another characteristic is that all of the BUs assigned to each territory are connected by a path contained enterely within the territory. In addition, the BUs in each territory must be relatively close to each other (compactness). One way to achieve this requirement is to minimize a dispersion measure. We use a dispersion measure based in the objective of the p-median problem (p-MP). All parameters are assumed to be known with certainty. We used a bi-objective optimization model introduced in [10]. In this model the compactness and the maximum deviation with respect to the number of customers are considered as objectives and the remaining requirements are treated as constraints. Let $N^i = \{j \in V : (i, j) \in E \vee (j, i) \in E\}$ be the set of adjacent nodes to node $i; i \in V$. The Euclidean distance between j and i is denoted by d_{ji}, $i, j \in V$. The average (target) value of attribute a can be computed as $\mu^{(a)} = w^{(a)}(V)/p$, $a \in A$.

Due to the discrete structure of the problem and to the unique assignment constraint, it is practically impossible to have perfectly balanced territories with respect to each attribute. Let $\tau^{(2)}$ be the specific tolerance allowed by the company to measure the relative deviation from average territory size with respect to sales volume.

2.2 Bi-objective Optimization Model

Decision variables

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

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

$$\text{Min } f_1 = \sum_{j \in V} \sum_{i \in V} d_{ji} x_{ji} \quad (1)$$

$$\text{Min } f_2 = \max_{i \in V} \left\{ \frac{1}{\mu^{(1)}} \left| \sum_{j \in V} (w_j^{(1)} x_{ji}) - \mu^{(1)} x_{ii} \right| \right\} \quad (2)$$

Subject to:

$$\sum_{i \in V} x_{ii} = p \quad (3)$$

$$\sum_{i \in V} x_{ji} = 1 \qquad \forall j \in V \tag{4}$$

$$\sum_{j \in V} w_j^{(2)} x_{ji} \geq (1 - \tau^{(2)})\mu^{(2)} x_{ii} \quad i \in V \tag{5}$$

$$\sum_{j \in V} w_j^{(2)} x_{ji} \leq (1 + \tau^{(2)})\mu^{(2)} x_{ii} \quad i \in V \tag{6}$$

$$\sum_{j \in \cup_{v \in S}(N^i \setminus S)} x_{ji} - \sum_{j \in S} x_{ji} \geq 1 - \mid S \mid \quad i \in V; S \subset [V \setminus (N^i \cup \{i\})] \tag{7}$$

$$x_{j,i} \in \{0,1\} \quad i,j \in V \tag{8}$$

Objective (1) represents a dispersion measure based on a p-MP objective. In this sense, minimizing dispersion is equivalent to maximizing compactness. The second objective (2) represents the maximum deviation with respect to the target size related to the number of customers. So, balanced territories should have small deviation with respect to the average number of customers. Constraint (3) guarantees the creation of exactly p territories. Constraints (4) guarantee that each node j is assigned to only one territory. Constraints (5)-(6) 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 $\tau^{(a)}$) around its average size. Constraints (7) guarantee the connectivity of the territories. Note that, as usual, there is an exponential number of such constraints.

3 Proposed GRASP Procedures

In general, GRASP is a metaheuristic that contains good features of both pure greedy algorithms and random construction procedures. 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 (7), 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 paper, we are introducing different GRASP schemes called BGRASP and TGRASP, 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: dispersion, maximun deviation with respect to the number of customers and total infeasibility in constraints (5) and (6). 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, maximum deviation, total infeasibility and total number of unconnected BUs. TGRASP-I and TGRASP-II are described in a very similar way to BGRASP-I and BGRASP-II, respectively. The only diference 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 (6). We described the GRASP strategies in a single scheme, see Procedure 1.

Procedure 1 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}) and the GRASP strategy (BGRASP-I, BGRASP-II, TGRASP-I or TGRASP-II) constitute the input. In order to explore the objective space in a best way. For each GRASP iteration a set of weights Λ is selected in such way that $\lambda \in \Lambda : \lambda \in [0, 1]$. 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 depends on the strategy chosen. Observe 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 (5)-(6) 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, in order to obtain feasible solutions, during the post-processing phase infeasibility is treated as the objective to be minimized. 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 our GRASP strategies, an approximation of the Pareto front is reported.

3.1 BGRASP Description

This strategy follows the generic scheme of GRASP (Procedure 1). A greedy function (9) during construction phase is a convex combination of two components weighted by λ which are related with the original objectives: dispersion measure (1) and maximum deviation (2). 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 Procedure 1 are detailed as follows.

Procedure 1 General scheme for BGRASP and TGRASP $(\alpha, iter_{max}, f, max_{moves}, strategy)$

INPUT: $(\alpha, iter_{max}, f, max_{moves}, strategy)$
α:= GRASP RCL quality parameter
$iter_{max}$:= GRASP iterations limit
f:= Minimum node degree required to create a subgraph which is used to select initial seeds in the ConstructSolution method
max_{moves}:= Maximum number of movements in the post-processing phase
strategy:= BGRASP-I, BGRASP-II, TGRASP-I or TGRASP-II
OUTPUT: D^{eff} set of efficients solutions
Λ: set of weights for greedy function selected in the range $[0, 1]$
$\Lambda \leftarrow$ generate (r); $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_r\}$ $D^{eff} \leftarrow \emptyset$
$D^{pot}(S) \leftarrow \emptyset$: set of potential efficients solutions
FOR$(l = 1, ..., iter_{max})$

```
    FOR EACH(λ ∈ Λ)

        IF(strategy is BGRASP-I) OR (strategy is TGRASP-I)
        Obj = 3 {Number of objectives to be optimized}
        ELSE
        Obj = 4
        END IF
        IF(strategy == BGRASP − I) OR (strategy == BGRASP − II)
            S ← ConstructSolutionBGRASP(α, f, λ, strategy)
        ELSE
            S ← ConstructSolutionTGRASP(α, f, λ, strategy)
        END IF
            FOR(g = 1, ..., Obj)
                D^pot(S) ←PostProcessing(S, max_moves, strategy, g, Obj)
                UpdateEfficientSolutions(D^eff, D^pot(S))
            END FOR
        END FOR

    END FOR

RETURN D^eff
```

3.1.1 BGRASP Construction Phase

In general, the construction phase consists of the assignment of BUs to territories keeping balanced territories with respect to the demand while seeking good objective values. Before the assignment process takes place p initial points are selected to open p territories, these points are the base for the assignment process. Previous work showed us 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 the territory stops way before reaching balancing. This implies some territories end up relatively small. So a better spread of the seeds is needed. In order to obtain best initial seeds we select p disperse initial points that have high connectivity degree. 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 [4] on G'. The p nodes are used as seeds to open p territories. Let $\{i_1, i_2, ...i_p\}$ be this set of disperse nodes. Then from this set, we start a partial solution $S = (V_1, V_2, ..., V_p)$ by setting $V_t = \{i_t\} \forall t \in \{1, 2., ...p\}$.

Then, at a given BGRASP construction iteration (see Procedure 2) we consider p partial territories and attempt 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, $c(t^*)$ is center of V_{t^*} and $N(V_{t^*})$ is the set of currently unassigned nodes that can be assigned to V_{t^*}. If $N(V_{t^*})$ is empty we take the next smallest territory and proceed iteratively. The cost of assigning a node j to territory V_{t^*} is given by the greedy function (9), this function weights the change produced in the objective values.

$$\phi(j, t^*) = \lambda f_{disp}(j, t^*) + (1 - \lambda) f_{dev}(j, t^*), \tag{9}$$

where

$$f_{disp}(j, t^*) = \frac{1}{d_{max}} \left(\sum_{i \in V_{t^*} \bigcup \{j\}} d_{ic(t^*)} \right) \tag{10}$$

$$f_{dev}(j, t^*) = \frac{1}{\mu^{(1)}} \max \left\{ w^{(1)} \left(V_{t^*} \bigcup \{j\} \right) - \mu^{(1)}, \mu^{(1)} - w^{(1)} \left(V_{t^*} \bigcup \{j\} \right) \right\} \tag{11}$$

and the normalization parameter

$$d_{max} = \frac{(|V| - p)}{p} \max_{i,j \in V} d_{ij} \tag{12}$$

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

$$\phi_{\min} = \min_{j \in N(t^*)} \phi(j, c(t^*)) \tag{13}$$

Procedure 2 ConstructSolutionBGRASP($\alpha, f, \lambda, strategy$)

INPUT: $(\alpha, f, \lambda, strategy)$
α:= GRASP RCL quality parameter
f:= Minimum node degree which is required to consider a node as an initial seed to open a new territory
λ:= 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** $c(t) \leftarrow i_t$; $V_t \leftarrow V_t \bigcup \{i_t\}$; $B \leftarrow B \setminus \{i_t\}$
WHILE$(B \neq \emptyset)$

$$l \leftarrow \arg \min_{t \in T: Flag(t)=1} \frac{w^{(2)}(V_t)}{\mu^{(2)}}$$

IF($strategy$ is BGRASP-I)

$$N(l) \leftarrow \bigcup_{i \in V_l} \{j \in N^i \text{ and } j \in B\} \ \{\textit{only connected nodes}\}$$

ELSE

$$N(l) \leftarrow B$$

END IF
IF $(N(l) \neq \emptyset)$

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 $k \in RCL$
$V_l \leftarrow V_l \bigcup \{k\}$; $B \leftarrow B \setminus \{k\}$
$c(l) \leftarrow \arg \min_{j \in V_l} \sum_{i \in V_l} d_{ji}$ $\{\textit{Update center}\}$

ELSE

$flag(t) \leftarrow 0\{\textit{Close territory}\}$

END IF

END WHILE
RETURN $S = (V_1, ..., V_p)$

$$\phi_{\max} = \max_{j \in N(t^*)} \phi(j, c(t^*)) \tag{14}$$

$$RCL = \{j \in N(t^*) : \phi(j, c(t^*)) \in [\phi_{\min}, \phi_{\min} + \alpha(\phi_{\max} - \phi_{\min})]\} \tag{15}$$

Then, a node i is randomly chosen from the RCL. We update the territory $V_{t^*} = V_{t^*} \bigcup \{i\}$ 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 we obtain a p-partition $S = (V_1, V_2, ..., V_p)$ that 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.

3.1.2 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). The motivation of this is that the search trajectories are well directed and it avoid the oscillation yielded by a multi-objective search. In addition, this local search has been applied successfully in SSPMO [7]. This process starts with the final solution obtained when the construction phase stops. Then, we start with a solution S (p-partition of V) such that $S = \{V_1, ..., V_p\}$. Additionally, $\forall V_t \in S$ a center $c(t) \in V_t$ is associated and $\forall i \in V_t$ a territory index $q(i) = t$ is known. S may be infeasible with respect to the balancing constraints (5) and (6), so in this phase BGRASP attempts to obtain feasible solutions and simultaneously it searches for solutions that represent the best compromise between the objective functions. In order to obtain feasible solutions during this phase, balancing constraints (5) and (6) are dropped and are considered as an additional objective function instead. In the case of BGRASP-I, there are three objectives that are minimized: (i) dispersion measure,(ii) maximum deviation with respect to the number of customers, and (iii) infeasibility related to the balancing of sales volume. 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 (19) and it computes the total number of unconnected nodes.

$$z_1(S) = \sum_{j \in V_t, t \in T} d_{jc(t)} \tag{16}$$

$$z_2(S) = \frac{1}{\mu^{(1)}} \max_{t \in T} \left\{ \max\{w^{(1)}(V_t) - \mu^{(1)}, \mu^{(1)} - w^{(1)}(V_t)\} \right\} \tag{17}$$

$$z_3(S) = \frac{1}{\mu^{(2)}} \sum_{t \in T} \max \left\{ w^{(2)}(V_t) - (1 + \tau^{(2)})\mu^{(2)}, (1 - \tau^{(2)})\mu^{(2)} - w^{(2)}(V_t), 0 \right\} \tag{18}$$

$$z_4(S) = |\eta(S)|, \tag{19}$$

where

$$\eta(S) = \bigcup_{t \in T} \{i \in V_t : \forall j \neq i, j \in V_t, (i, j) \notin E\} \tag{20}$$

The Post-processing phase attempts to find potential efficient solutions in the neighborhood of S. For doing that, we define a neighborhood $N(S)$ which is the solutions 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, Procedure 4 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 we select the $move(i, j)$, the neighbor solution $\overline{S}$ is given by $\overline{S} = (..., V_{q(i)} \setminus \{i\}, ..., V_{q(j)} \bigcup\{i\}, ...)$. The $move(i, j)$ is accepted only if this improves the value of the objective function that is being optimized in that moment (see Procedure 4).

Procedure 3 PostProcessing($S_0, iter_{max}, g, Obj$)

INPUT $(S_0, iter_{max}, g, Obj)$
$S = S_0$:= {*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** $(count < iter_{max})$
> $(i, j) \leftarrow$ select_move$(N(S))$
> $N(S) \leftarrow N(S) \setminus \{(i, j)\}$
> $acceptable \leftarrow$ evaluate_move$(S, (i, j), z_g)$
> **IF** $(acceptable)$
>
> > $S_{t(i)} \leftarrow S_{t(i)} \setminus \{i\}$
> > $S_{t(i)} \leftarrow S_{t(j)} \bigcup\{i\}$
> > $count \leftarrow count + 1$
> > update$(N(S))$
> > **IF** (IsFeasible$(S) == YES$)
> > update_NDS(D, S)
> > **END IF**
>
> **END IF**
> **END WHILE**
> **IF**$(h < Obj)$
> $h = h + 1$
> **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 [2] and used by Molina, Martí, and Caballero [7]. The linking of single-objective local search schemes is made considering different ordering of the objective functions being

Procedure 4 evaluate_move($S, (i, j), g$)

INPUT: $(S, (i, j), g)$
S:= Current solution
(i, j):= Intended move
g:= Objective function index that should be optimized
OUTPUT: TRUE if (i, j) is acceptable FALSE otherwise
$\bar{S} \leftarrow S : S_{t(i)} \setminus \{i\}, S_{t(j)} \bigcup \{i\}$ {*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($z_4(S) == 0$) {*S is a connected solution*}

IF($z_g \neq z_4$)

IF($\triangle z_g > 0$) **AND** ($z_4(\bar{S}) == 0$) **RETURN** TRUE
ELSE RETURN FALSE
END IF

ELSE

IF($\triangle z_1 || \triangle z_2 || \triangle z_3$) **AND** ($z_4(\bar{S}) == 0$) **RETURN** TRUE
ELSE RETURN FALSE
END IF

END IF

ELSE

IF($\triangle z_g > 0$) **RETURN** TRUE
ELSE RETURN FALSE
END IF

END IF

pursued. Suppose we select the optimization order as $(z_1(S), z_2(S), z_3(S))$, then the local search path is as follows: The first local search starts with S a final solution after the construction phase and attempts to find the optimal solutions to the problem with the single objective $z_1(S)$ (16). Let S^1 be the best point visited at the end of this search. Then a local search is applied again to find the best solution to the problem with the single objective $z_2(S)$ (17) 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 $z_3(S)$ (18) and the initial solution S^2 obtained in the previous optimization. At this point, we solve again the problem with the first objective $z_1(S)$ starting from S^3. This phase yields at least 3 points that approximate the best solutions to the single objective problems that result from ignoring all except one objective function. During this phase only feasible solutions are kept and a potential set of nondominated solutions are kept (see Procedure 5), too. Additionally, efficient solutions may be found because all potential nondominated solutions are checked for inclusion in the efficient set E (see Procedure 6). This efficient set E is updated according to Pareto efficiency, this check is made over the original objectives: dispersion (16) and maximum deviation with respect to the number of customers (18) (see Algorithms 5 and 6).

Procedure 5 update_NDS(D, S)

INPUT: (D, S)
OUTPUT: D *set of nondominated solutions*
$eff \leftarrow 1$:= 1 if the solution S is efficient, 0 in otherwise
FOR ALL $S' \in D$

  IF$((z_1(S) \geq z_1(S'))$ **AND** $(z_2(S) \geq z_2(S')))$
  $eff \leftarrow 0$
  END IF

END FOR
IF(eff)

  FOR ALL $S' \in D$ **DO**

    IF$((z_1(S) \leq z_1(S'))$ **AND** $(z_2(S) \leq z_2(S')))$
    $D \leftarrow D \setminus \{S'\}$
    END IF

  END FOR
  $D \leftarrow D \bigcup \{S\}$

END IF
RETURN D

Procedure 6 UpdateEfficientSlutions(D^{eff}, D^{pot})

INPUT: (D^{eff}, D^{pot})
D^{eff}:= Set of current efficient solutions
D^{pot}:= Set of potential nondominated solutions
OUTPUT: D^{eff} Efficient set
FOR ALL $S \in D^{pot}$ **DO** update_NDS(D^{eff}, S)
RETURN D^{eff}

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 $f_j(x) < f_j(x^*)$. So, in our case $g = 2$.

We can repeat linking process local searches using a different ordering of the objectives. In this work, we explored different trajectories depending on the number of objectives to be optimized. For instance, in BGRASP we used the following trajectories that 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.

3.2 TGRASP Description

These procedures 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 Procedure 7). During this phase the greedy function (21) is a convex combination (23) of three components: dispersion measure (10), maximum deviation (11) and maximum infeasibility with respect to the upper bound of sales volume balancing (22). 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 a greedy function (21).

$$\gamma(j,t) = \lambda_1 f_{disp}(j,t) + \lambda_2 f_{dev}(j,t) + \lambda_3 f_{infeas}(j,t) \tag{21}$$

Where

$$f_{infeas}(j,t) = \frac{1}{\mu^{(2)}} \max\left\{(1+\tau^{(2)})\mu^{(2)} - w^{(2)}\left(V_t \bigcup \{j\}\right), 0\right\} \tag{22}$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \tag{23}$$

Note that (22) penalize only those assignments that make infeasible the balancing constraint given by (6). The post-processing phase of TGRASP procedures is the same as in BGRASP strategies (see Procedure 3). Note that, in the TGRASP-I and TGRASP-II, during the local search, four objectives are minimized: (i) dispersion measure (16), (ii) maximum deviation with respect to the number of customers (17), (iii) infeasibility related with balancing of sales volume (18), and (iv) total number of unconnected nodes (19). The updating of efficient solutions is made considering only feasible solutions.

4 Experimental Results

An evaluation of the diverse strategies proposed in this work is carried out. We generate two instance sets with $(n, p) \in \{(1000, 50), (500, 20)\}$. We randomly generated 10 instances based on real-world data provided by the industrial partner for each set. We used $\tau^{(2)} = 0.05$ and 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 were 2020 and 2000 was the maximum number of movements during the post-processing phase. During our experimental work, we observed 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, we made a filtering of solutions in order to apply the post-processing phase only over a set of the best solutions which were evaluated according to the merit function given by (24). We tested other merit functions that empirically showed poor behavior. That motivated the use of this function for filtering solutions. Note that, each component is normalized.

Procedure 7 ConstructSolutionTGRASP($\alpha, f, \lambda, strategy$)

INPUT: $(\alpha, f, \lambda, strategy)$
α:= GRASP RCL quality parameter
f:= Minimum node degree which is required to consider a node as an initial seed to open a new territory
λ:= 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** $c(t) \leftarrow i_t$; $V_t \leftarrow V_t \bigcup \{i_t\}$; $B \leftarrow B \setminus \{i_t\}$
WHILE$(B \neq \emptyset)$

$$l \leftarrow \arg\min_{t \in T} \frac{w^{(2)}(V_t)}{\mu^{(2)}}$$

IF($strategy$ is TGRASP-I)

$$N(l) \leftarrow \bigcup_{i \in V_l} \{j \in N^i \text{ and } j \in B\} \; \{\textit{only connected nodes}\}$$

ELSE

$$N(l) \leftarrow \bigcup \{j \in B\} \; \{\textit{any node can be assigned}\}$$

END IF
IF $(N(l) \neq \emptyset)$

ComputeGreedyFunction $\gamma(j, c(l))$ **FOR ALL** $j \in N(l)$
$\gamma_{min} \leftarrow \min_{j \in N(l)} \gamma(j, c(l))$; $\gamma_{max} \leftarrow \max_{j \in N(l)} \gamma(j, c(l))$
$RCL \leftarrow \{j \in N(l) : \gamma(j, c(l)) \in [\gamma_{min}, \alpha(\gamma_{max} - \gamma_{min})]\}$
Random selection of $k \in RCL$
$V_l \leftarrow V_l \bigcup \{k\}$; $B \leftarrow B \setminus \{k\}$
$c(l) \leftarrow \arg\min_{j \in V_l} \sum_{i \in V_l} d_{ji}$ $\{\textit{Update center}\}$

ELSE

$flag(t) \leftarrow 0\{\textit{Close territory}\}$

END IF

END WHILE
RETURN $S = (V_1, ..., V_p)$

$$\rho(S) = \frac{2f_{disp}(S)}{(|V|-p)d_{Max}} + \frac{f^{(1)}_{infeas}}{p} \tag{24}$$

Where,

$$f_{disp}(S) = \sum_{t \in T} d_{jc(t)} : j \in V_t \tag{25}$$

$$f^{(1)}_{infeas} = \sum_{t \in T} \left\{ \frac{1}{\mu^{(1)}} \max \left\{ (1-\tau^{(1)})\mu^{(1)} + w^{(2)}\left(V_t\right), (1+\tau^{(1)})\mu^{(1)} - w^{(2)}\left(V_t\right), 0 \right\} \right\} \tag{26}$$

$$d_{Max} = \max_{i,j \in V} d_{ij} \tag{27}$$

We selected 100 (out of 2020) solutions in such a way that these solutions have the smallest values in the merit function given by (24). The post-processing phase (described in Procedure 3) was applied over the set of these filtered solutions.

We carried out an experimental work based on a factorial design with two factors (called *strategy* and *type*, respectively). We considered two levels for each of them, *strategy* $\in \{BGRASP, TGRASP\}$ and *type* $\in \{I, II\}$ and for each combination of factors we tested 10 replicates. Figures 1 and 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 1 and Figure 2, respectively. So, 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 main goal in the first part of our experimental work was to analyze the effects produced by each factor over a set of standard performance measures used in multi-objective optimization. The performance measures that we employed 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 [14] in connection with the computational testing of SPEA2 is based on the kth-nearest neighbor method of Silverman [12]. This metric is simply the distance to the kth-nearest efficient point. We use k=4 and calculate both the mean and the max of kth-nearest distance values. So, the smaller the k-distance the better in terms of the frontier density.

3. *Size of space covered (SSC):* This metric suggested by Zitzler and Thiele [15]. This measure compute the volume of the dominated points. Hence, larger the value of SSC the better.

4. *C(A,B):* It is known as the coverage of two sets measure [15]. This measure represents the proportion of points in the estimated efficient B that are dominated by the efficient points in the estimated frontier A.

Tables 1 and 2 contain a summary of different performance measures for instances from $(500, 20)$ and $(1000, 50)$, respectively. We carried out an ANOVA for each performance measure, these analyses were based on a general linear model with interaction of factors. In those 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 3 contains a summary of P related to estimated effects over different performance measures. Suppose that we consider a significance level $\alpha = 0.05$ during the significance testing, P-values for instances set $(1000, 50)$ show the SSC measure is very sensitive to any change in the individual factors and in any

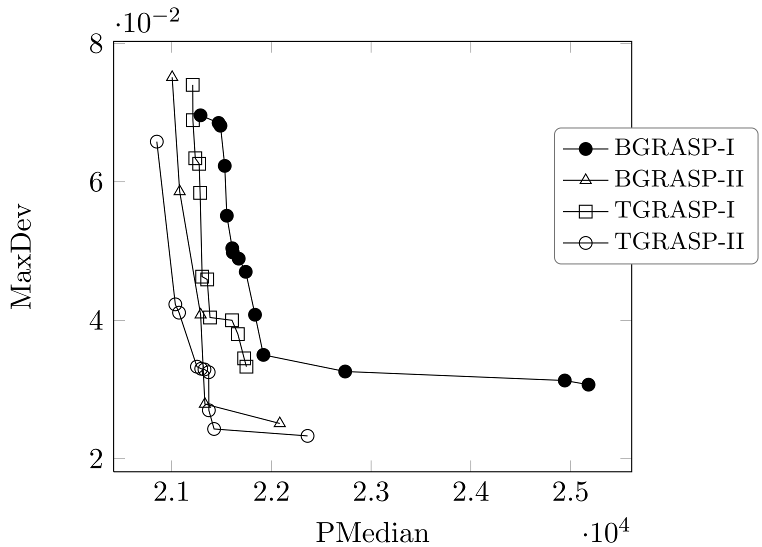

Figure 1: Efficient frontiers for instance ds_500_20-03

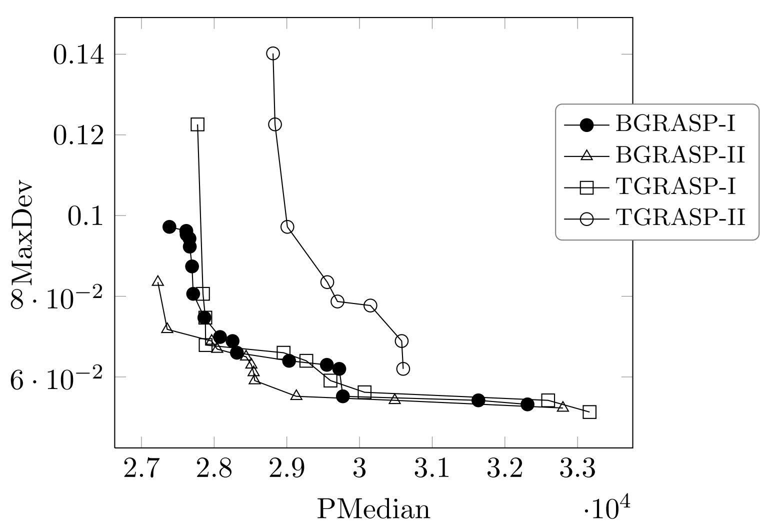

Figure 2: Efficient frontiers for instance ds_1000_50-08

interaction between these. In contrast, when the performance measures are the *k-distance(mean)* or number of points, we 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. It is, the way to consider the connectivity requirement affects this performance measure. Remember 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 objective function 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. So, in the ANOVA analyses SSC is the only performance measure that presents significant variation generated from the factor levels.

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

Table 1: Summary of metrics used during ANOVA, instances (500,20)

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

Table 2: Summary of metrics used during ANOVA, instances (1000,50)

Instance $(1000, 50)$				
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)				
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
Term	*k-distance(mean)*	*k-distance(max)*	N. of points	SCC

Table 3: Summary of P-values associated to estimated effects from factors to performance measures

Figures 3 and 4 show the mean values of factors interaction for SSC measure. Remember that, hight values of SSC are better than small values, so, for instances from $(500, 20)$ (see Figure 3) TGRASP-I and

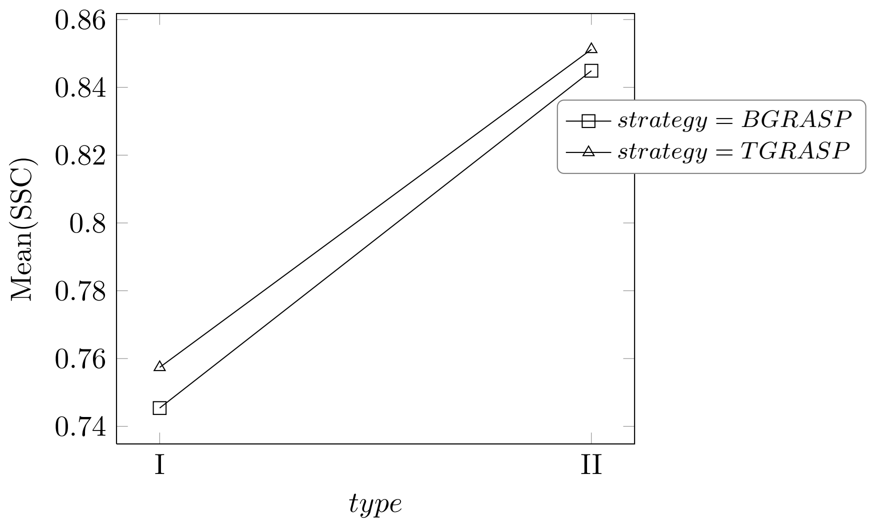

Figure 3: Interaction plot for SSC measure, instances $(500, 20)$

TGRASP-II are lightly better than BGRASP-I and BGRASP-II, respectively. In contrast, for instances from $(1000, 50)$ (see Figure 4) TGRASP-II obtained the 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 4 and 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 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 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.

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

Table 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	0.328	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

Table 5: Mean value of coverage of two sets measure for instances from $(1000, 50)$

The last important feature to be evaluated is the optimization time required for each GRASP procedure. For instances from $(500, 20)$ (see Figure 6), the best time is for those procedures that keep the connectivity

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

Table 7: Time (seconds) for instances from $(1000, 50)$

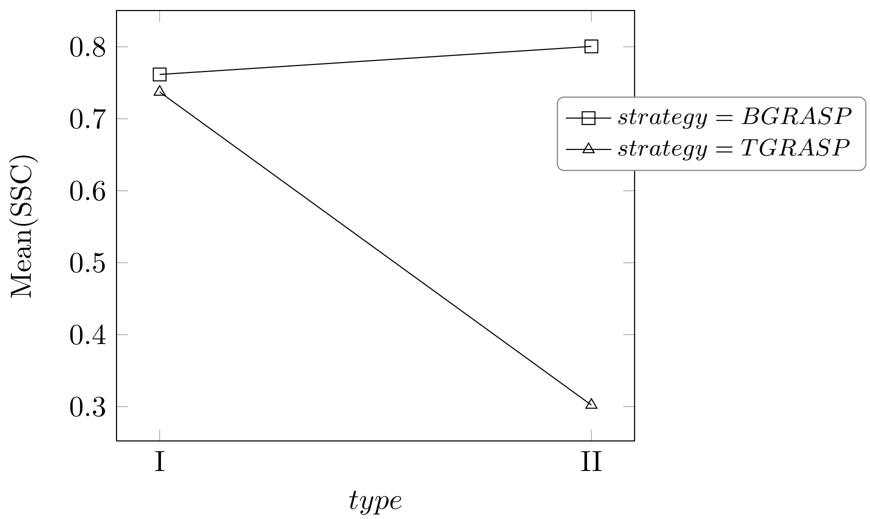

Figure 4: Interaction plot for SSC measure, instances $(1000, 50)$

5 Conclusions

In this paper we have presented two GRASP procedures called BGRASP and TGRASP for a bi-objective territory design problem with connectivity and balancing constraints. The problem arises from a real-world situation in a beverage distribution company in Monterrey, Mexico. Two variations of each GRASP procedure are evaluated.

We carried out an evaluation of these procedures based on well-known performance measures used in multi-objective optimization. These measures are: number of points, size of the space cover (SSC), k-distance, coverage of two sets measure, and optimization time. 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.

An ANOVA was carried out for each of the following performance measures: number of points, k-distance (mean), k-distance (max), and SSC. We observed that only the SSC measure presents significant variation yielded by the applied GRASP procedure, the worst behavior was for TGRASP-II. In contrast, the number of points and k-distance did not have significant changes independently to the used GRASP procedure.

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

In summary, we presented four solution procedures for a bi-objective territory design problem, these are good alternatives to the decision maker, the choice of one of them depends on the performance measure that is important to the decision maker. In this work we analyzed the standard measures used in multi-objective optimization.

Acknowledgements: This work was supported by 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. The research of the first author has been funded by a doctoral fellowship from Universidad Autónoma de Nuevo León, grant NL-2006-C09-32652.

The authors wish to thank Professor Julián Molina from Universidad de Málaga, Spain, for his help for computing some performance measures.

References

[1] 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.

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

[3] 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.

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

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

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

[7] 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.

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

[9] 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.

[10] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and J. L. González-Velarde. Constructing efficient commercial territory design plans for a beverage distribution firm with a bi-objective programming model. Technical Report PISIS-2009-08, Graduate Program in Systems Engineering, Universidad Autónoma de Nuevo León, San Nicolás de los Garza, Mexico, December 2009.

[11] 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.

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

[13] 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.

[14] 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.

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

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