A GRASP with Strategic Oscillation for a Commercial Territory Design Problem with a Routing Budget Constraint

Roger Z. Ríos-Mercado[1] and Juan C. Salazar-Acosta[2]

[1] Universidad Autónoma de Nuevo León, Graduate Program in Systems Engineering, AP 111–F, Cd. Universitaria, San Nicolás de los Garza, NL 66450, Mexico
roger.rios@uanl.edu.mx
[2] Alpha5, Monterrey NL, Mexico,
juan.salazar@aleph5.com

Abstract. This paper addresses a commercial districting problem arising in the bottled beverage distribution industry. The problem consists of grouping a set of city blocks into territories so as to maximize territory compactness. As planning requirements, the grouping seeks to balance both number of customers and product demand across territories, maintain connectivity of territories, and limit the total cost of routing. A combinatorial optimization model for this problem is introduced. Work on commercial territory design has particularly focus on design decisions. This work is, to the best of our knowledge, the first to address both design and routing decisions simultaneously by considering a budget constraint on the total routing cost in commercial territory design. A greedy randomized adaptive search procedure (GRASP) that incorporates advanced features such as adaptive memory and strategic oscillation is developed. Empirical evidence over a wide set of randomly generated instances based on real-world data show a very positive impact of these advanced components. It was observed how these strategies yield feasible solutions, which is something hard to achieve when these componentes are not considered. Solution quality is significantly improved as well.

Keywords: Territory design; Routing cost; GRASP; Strategic oscillation; Adaptive memory

1 Introduction

Territory design consists of grouping small geographical units into larger clusters called territories for the purpose of making more manageable the entire set. Current applications are vast: political and school districting (Hojati [13], Bozkaya, Erkut, and Laporte [4]), sales and commercial design (Kalcsics et al. [14], Zoltners and Sinha [20]), health-care districting (Pezzella and Nicoletti [17]), school districting (Caro et al. [6]), emergency services (Bertolazzi, L. Bianco, and Ricciardelli [3]), salt spreading operations (Muyldermans et al. [16]), recollection of waste electric and electronic equipment (Fernández et al. [9]), and electrical power districting (Bergey, Ragsdale, and Hoskote [2]), to name a few. A more comprehensive survey on territory design can be found in the work of Kalcsics, Nickel, and M. Schröder [15], Duque, Ramos, and Suriñach [7], and Zoltners and Sinha [21].

This paper deals with a commercial territory design problem (CTDP) motivated by a real-world application in a beverage distribution firm. The firm seeks to form territories that are as compact as possible subject to planning criteria such as territory balance with respect to two different activities (number of customers and product demand), territory connectivity, and unique assignment.

The CTDP problem was introduced by Ríos-Mercado and Fernández [18]. Other variations of this problem have been studied recently [5, 19]. In each of these works, a dispersion measure for assessing territory compactness based on Euclidean distances is considered. In this regard, a good or desirable districting plan is one with low values of this dispersion measure. This is similar to certain location problems such as the p-Median Problem and the p-Center Problem where a dispersion based measure is minimized. Now, it is clear than in real world applications, network-based distances are more representative of distances between basic units. In this regard, in many cases one can use models based on Euclidean distances without loss of generality because one can always replace Euclidean distances by their corresponding shortest path distances. This is also true in TDP applications where connectivity constraints are not considered. When connectivity constraints are taken into account, the fact that the territory dispersion measure is limited to those paths entirely contained in the territory leads to very intractable models as the distances between basic units end up being solution-dependent. Therefore, in previous work on TDPs with connectivit constraints authors have address the problems under the assumption of the Euclidean-based distances to make the problems more tractable. It is easy to find examples where a shortest path between two nodes in a given territory falls outside the territory. In addition, a single budget constraint on the total routing cost is introduced in our problem. These two aspects (network-based distances and routing cost) have been often neglected in previous work on commercial territory design and it represents a more challenging problem. Among the few works addressing the design and routing simultaneously is the one of Haugland, Ho, and Laporte [12] but in a different application. To the best of our knowledge, our work is the first to address these issues in this class of TDPs.

To address this problem, a Greedy Randomzied Adaptive Search Procedure (GRASP) with adaptive memory and strategic oscillation is proposed and evaluated over a range of randomly generated instances based on real-world data. The iterative procedure is formed by three steps (two in a construction phase and one in an improvement phase). First a partial territory design, where not all units have been assigned to territories, is built by taking into account the dispersion-based objective function. In the second step, the remaining unassigned units are assigned to territories by using a merit function that weights both the objective function and the violation of the balancing constraints. This merit function includes a frequency-based term as an adaptive memory mechanism. In this construction phase the routing constraint is relaxed. After a solution is built, routing costs are evaluated by solving a small-scale Traveling Salesman Problem with Multiple Visits (TSPM) [11] within each territory including the distribution center. In the post-processing phase, the procedure attempts to improve the objective function and to satisfy the violated balancing and routing constraints. To this end, strategic oscillation is performed. Strategic oscillation is a very powerful technique [10] that consists of relaxing some constraints and add them to the objective function with a penalty parameter. This penalty parameter is dynamically updated throughout the execution of the algorithm This dynamic update allows the search trajectory to oscillate between the feasible and the infeasible space (those solutions not satisfying these relaxed constraints). The motivation for this comes from the fact that, by allowing the problem to become temporarily unfeasible, it is possible to visit solution regions that otherwise would be impossible to explore. The results are very encouraging. Empirical evidence indicates the great impact of the proposed advanced components, particularly the use of strategic oscillation within the local search.

The rest of the paper is organized as follows. In Section 2, the problem description and its corresponding combinatoriaol model are presented. Section 3 describes the proposed approach. The empirical work is discussed in Section 4. Final remarks are drawn in Section 5.

2 Problem Description

Each node or basic unit (BU) represents a city block which might contain one or more customers. Every edge represents adjacency between two BUs. Let $G = (V, E)$ be an undirected and planar graph that represents the distribution network, where V is the set of all BUs (nodes) and E is the set of edges, representing the connectivity between BUs. Let d_{ij} be the Euclidean distance between BUs i and j, $(i, j) \in E$. The subset $X_k \subseteq V$ represents a territory. Every node i has two different activities associated to it: number of customers (measured by parameter w_i^1) and total demand (measured by parameter w_i^2). The size of territory X_k with respect to activity a is given as follows: $w^a(X_k) = \sum_{i \in X_k} w_i^a$, $a = 1, 2$. The perfect measure of a territory with respect to activity a is defined by $\mu^a = w^a(V)/p$, where p is the known number of territories. Because of the discrete nature of the problem, it is almost impossible to obtain this perfect measure for each activity. To cope with this, the company allows a tolerance deviation (represented by τ^a) from the average value μ^a. Let T_{ij}^S be the shortest path from node i to node j in $G^S = (S, E(S))$, the subgraph of G induced by set S, $S \subseteq V$, $i, j \in V$ with corresponding length t_{ij}^S. To model the dispersion of a given design $X = (X_1, \ldots, X_p)$ we use the diameter-based measure given by:

$$f(X_1, X_2, \ldots, X_p) = \max_{k=1,\ldots,p} \left\{ \max_{i,j \in X_k} \left\{ t_{ij}^{X_k} \right\} \right\}$$

It is required that each territory be connected, i.e., one should be able to traverse a single territory without having to pass through other territories.

Let Π be the collection of all possible p-partitions of V, and let $X = (X_1, \ldots, X_p)$ be an arbitrary partition $X \in \Pi$. The combinatorial optimization TDP model is shown below:

$$\min_{X \in \Pi} f(X) = \max_{k=1:p} \left\{ \max_{ij \in X_k} \left\{ t_{ij}^{X_k} \right\} \right\} \tag{1}$$

subject to

$$w^a(X_k) \in \left[(1-\tau^a)\mu^a, (1+\tau^a)\mu^a\right], \; a = 1, 2 \tag{2}$$

$$G^{X_k} = (X_k, E(X_k)) \text{ must be connected, } k = 1, 2, \ldots, p \tag{3}$$

$$\sum_{k=1,\ldots,p} R(X_k) \leq C \tag{4}$$

In the objective we use a diameter-based function (1) to measure territory dispersion. Constraints (2) assure that each territory is appropriately balanced. The requirement of territory connectivity is represented by (3), which means that, for every territory X_k, there exists a path from every node in the territory to every other node in the territory. Constraint (4) imposes a limit (denoted by C) on the total traveling cost, where $R(X_k)$ represents the routing cost of territory X_k. Assuming the distribution center is represented by node 0, $R(X_k)$ can be seen as the routing cost incurred when traversing the nodes in X_k from the distribution center and returing to it. Now, in the real-world problem, each territory routing is responsability of one truck. As $G = (V, E)$ represents a planar graph, we cannot assure the existence of a Hamiltonian cycle within each territory. In fact, what the company simply needs is that each basic unit is visited at least once, that is, if in a given routing a node is visited more than once, the truck makes the stop to deliver the product in the first visit only. The problem of finding a closed walk of minimum length where each node is visited at least once is known as the Traveling Salesman Problem with Multiple Visits (TSPM) [11]. This problem can be transformed into a TSP by replacing the edge cost with the shortest path distances in G. In the absence of negative cycles, such as our problem, shortest path distances between all pairs of nodes of a graph can be computed using efficient algorithms. Thus, $R(X_k) = \text{TSPM}\ (X_k \cup \{0\}) = \text{TSP}(X_k \cup \{0\})$, where the TSP operates in a graph that uses shortest path distances $t_{ij}^{X_k}$ instead of d_{ij}. Given that in our particular problem, each individual territory has approximately 25-40 BUs in the worst case, computing $R(X_k)$ is reduced to solving a TSP with 30-40 cities. This can be computed exactly very efficiently by state-of-the-art branch-and-cut methods. In our case, we use CONCORDE [1].

3 Proposed Algorithm

GRASP [8] is a metaheuristic that has been widely used to solve a large number of combinatorial optimization problems. In each GRASP iteration there are two phases: construction and post-processing. Construction phase aims at building a feasible solution by combining both greedy heuristics and randomization in a way that allows to construct a diverse number of good quality solutions. The post-processing attempts to improve the solution obtained in the first phase by means of local search.

Our GRASP construction phase has two stages. First a partial solution with p territories is built by using a GRASP greedy function that considers the objective function only. Then the remaining units are assigned to the territories by using a greedy function that incorporates some penalty terms associated to the violation of the balancing constraints. The post-processing phase consists of a local search scheme in which a node is selected to be moved from one territory to another. Both phases are described next.

3.1 Construction Phase

In this phase, the routing constraint is relaxed. After the solution is built, this constraint is taken into account in the local search phase.

We attempt to build p territories one at a time. We first select one node to be the seed and then construct the territory by adding nodes (using a greedy function based only on a dispersion term) until a closing criteria is met. To start the next territory, we then select the minmax node, that is, the node whose minimum distance with respect to the territories constructed is the largest, and start adding nodes again until the same closing criteria is met. This is done until p territories are formed.

Depending on how tight or loose this user-defined "closing" criteria is, there may be still many unassigned nodes at the end of this step. The second step consists of assigning the remaining nodes by following an Adaptive Memory Programming scheme. In this step a merit function consisting of the sum of the original function, the violation of the balancing constraints, and a frequency-based memory term, is used.

Thus, for the first stage of the construction phase, let X_k be the partial territory being formed, and let N be the set of nodes adjacent to X_k, that is, the set of nodes that do not belong to X_k but that share an edge with a node in X_k. N is called the candidate list. For every $v \in N$ we evaluate the greedy function

$$\phi(v) = \max \left\{ f_k(X_k), \max_{j \in X_k} \{t_{vj}\} \right\},$$

where $f_k(X_k)$ is the contribution to the objective function of the k-th territory. The idea behind GRASP is to construct a restricted candidate list (RCL) containing the best moves. To do this a quality threshold parameter α is defined and the RCL is formed by those elements in N such that their corresponding greedy function evalution falls within α percent of the best move. That is RCL $= \{v \in N \ : \ \phi(v) \leq \Phi^{\min} + \alpha(\Phi^{\max} - \Phi^{\min})\}$, where $\alpha \in [0,1]$, $\Phi^{\min} = \min_{v \in N}\{\phi(v)\}$ and $\Phi^{\max} = \max_{v \in N}\{\phi(v)\}$. By defining the α this way, it is clear that a value of $\alpha = 0$ corresponds to a purely deterministic greedy approach, and $\alpha = 1$ corresponds to a purely randomized approach. We randomly choose one node from the RCL and we add it to V_k. In addition to dispersion, we also wish to have balanced territories according to both activities. As we are adding nodes to each territory, we are interested in not violating the upper bound of the balance constraints. Thus, if there is $a \in \{1,2\}$ such that $w^a(X_k) > \beta(1+\tau^a)\mu^a$ the territory is closed. β is a positive parameter that allows the user to control how early the territory must be closed. The motivation for introducing this parameter stems from the observation that allowing the territory to close earlier may give more flexibility for the remaining unassigned units to be assigned to different territories.

In Step 2, we use the greedy function

$$\varphi(v) = \lambda\phi(v) + (1-\lambda)G_k(v) + \vartheta(v),$$

where $\lambda \in [0,1]$, and $\phi(v)$ is the same greedy function used in the previous step,

$$G_k(v) = \sum_a (1/\mu^a) \max\{w^a(X_k \cup \{v\}) - (1+\tau^a)\mu^a, 0\}$$

represents the sum of relative infeasibilities with respect to the upper bound of the balance constraints and $\vartheta(v)$ is the average number of times node v ended up in the same territory than the rest of the nodes in the current territory, that is $\vartheta(v) = \sum_{i \in X_k} frec(i,v)/(|X_k| iter)$, where $frec(i,v)$ tallies the number of times nodes i and v have belonged to the same territory in the past $iter$ GRASP iterations. $iter$ is used as normalization factor. This represents the adaptive memory component. Note that in this construction phase we do not consider the violation with respect to the lower bound of the balancing constraints because when we start a territory from zero and we iteratively add nodes, the lower bound of the balance constraints never gets worse. This is handled in the post-processing phase.

Once the construction phase ends, we need to evaluate the routing costs. As stated before, solving the TSPM is equivalent to solving a TSP, so this is practicaly reduced to solving p TSPs, one for each territory. It is well-known that nowadays one can solve relatively large TSP's by branch-and-cut methods. In our case, the size of each of the individual TSPs to be solved for computing these costs is no more than 30 to 40 nodes. This implies one can use a branch-and-cut method for optimally solving each TSP in a relatively short amount of time. In this specific case, we use CONCORDE [1] to solve the corresponding TSP within each built territory. The connectivity requirement is kept during the entire procedure.

3.2 Post-processing Phase

The aim of the local search is to improve the objective function and at the same time to reduce the infeasibilities of the balance and the routing constraints as much as possible.

The local search considers a neighborhood $N(S)$ of a partition S that consists of all possible movements of node i from its current territory $t(i)$ to the territory of another adjacent node j, $t(j)$, such that $(i,j) \in E$ and $t(i) \neq t(j)$.

We use a merit function with three terms. This merit function measures the objective function, the infeasibility of the budget constraint and the infeasibility of the balance constraints. This merit function is defined for a given partition $S = (X_1, \ldots, X_p)$ as follows:

$$\psi(S) = F(S) + \sigma H(S) + \gamma G(S) \tag{5}$$

where

$$F(S) = \max_k \left\{ \max_{ij \in X_k} \left\{ p_{ij}^{X_k} \right\} \right\}$$

is the original dispersion measure (diameter),

$$H(S) = \max \left\{ \sum_k R(X_k) - C, 0 \right\}$$

is the relaxed budget constraint, and

$$G(S) = \sum_k \sum_{a \in A} g^a(X_k)$$

is the sum of all relative infeasibilities of the balance constraints, with

$$g^a = \left(\frac{1}{\mu^a} \right) \max \left\{ w^a(X_k) - (1 + \tau^a)\mu^a, (1 + \tau^a)\mu^a - w^a(X_k), 0 \right\},$$

and σ and γ are penalty parameters to be dynamically updated as explained below.

Strategic oscillation: The parameters σ and γ in (5) are self-adjustable according to strategic oscillation [10]. When the budget constraint is violated, σ doubles its value and the same occurs with γ when the balance constraints are violated. When we have a feasible solution, both parameters reduce their values by half. With this strategy we can guide the search to a larger space by allowing infeasible moves. This technique has proven successfull in many combinatorial optimization problems, particularly in some territory design applications. For instance, Bozkaya, Erkut, and Laporte [4] make use of this idea for successfully handling some difficult constraints in a political districting problem. As it will be seen in the following section, this strategy gave very good results in our case as well.

4 Empirical Work

Our heuristic was coded in C++ and compiled with the GNU g++ version, under the Ubuntu Linux 9.10 OS in a computer with an Intel Processor(R) Core(TM)2 Quad CPU Q6600 of 2.40 GHz.

We used two types of instances: DU05 and DU10. Both types were taken from the database of Ríos-Mercado and Fernández [18]. DU05 instances have a balance deviation parameter $\tau = 0.05$ and DU10 have a $\tau = 0.10$. The budget limit C is given by the firm with a value of 20,000 for both types of instances. The size of these insatnces is of $n = 1000$ and $p = 40$.

4.1 Evaluation of the Effect of β

The β parameter is the one that controls the size of the territories being formed during the construction phase. A large (low) value of β implies a large (low) value of the territory size. The idea is to assess the trade-off between the degree of infeasibility obtained in the construction phase and the computational cost. The introduction of this parameter was based on the observation that trying to hit the normal upper bound as closing criteria (which corresponds to $\beta = 1$) produced many infeasible solutions. What happens is that when the method is building the territories, a large value of β implies that the constructed territories are relatively large, leaving very few choices of nodes to be assigned in the second step of the construction. These nodes are often located in places where the only available choice for assignemnt is a bad one. By closing the territories earlier (decreasing β) the constructed partial territories are smaller, and so more choices for the remaining nodes can be taken. To accomplish this, we tried different values of $\beta \in \{0.4, 0.5, 0.6, 0.7, 0.8\}$. We set our GRASP algorithm with an $\alpha = 0.1$, which was the best value obtained in an earlier experiment, and an iteration limit of 500. We also set the value of $\lambda = 0.2$ (the greedy function prameter).

The deviation from best solution (DFB) is computed as

$$\text{DFB} = 100 \times \frac{(f_\beta - f_{best})}{f_{best}}.$$

We have basically two types of constraints: the balance and the budget constraints. In the following tables we are comparing not only the average objectives, but also the degree of infeasibility of these constraints.

Table 1. Sensitivity analysis of β on DU10 and DU05 instances.

		β				
		0.4	0.5	0.6	0.7	0.8
DU10	DFB	14.4	13.5	9.4	2.1	(*)
	RCI	0.0	0.0	0.0	0.0	(*)
	BCI	0.1	0.0	0.0	0.0	(*)
DU05	DFB	16.2	16.9	12.6	2.4	(*)
	RCI	3.6	0.0	0.0	0.0	(*)
	BCI	0.0	0.0	0.0	0.0	(*)

(*) All infeasible

Table 1 shows the average results for DU10 and DU05 instances. The third row shows the average deviation from best objective (DFB) for DU10. The fourth and fifth row show the average relative infeasibility with respect to the budget (RCI) and balancing constraints (BCI), respectively. BCI is the sum of all the relative infeasibilities of each balancing constraints, which in turn is computed as the absolute violation of the balancing constraints divided by the corresponding average target size μ^a. For the budget constraint, RCI is computed as the absolute violation of the balancing constraint divided by the value of the upper bound C. Rows 6 to 8 show the results for DU05 instances.

As it can be seen, using a value of $\beta = 0.8$ makes the method fail on findining feasible solutions, as explained before. Then, as β decreasees (β = 0.6, 0.7), the method indeed improves to the point on finding feasible solutions for all instances tested. As β gets smaller, a trade-off is incurred, and the method starts failing for some instances. In terms of evaluating the objective function, using $\beta = 0.7$ gives better solutions on average (2.1 and 2.4 % deviation from best for DU10 and DU05, respectively). Therefore, $\beta = 0.7$ was consistently found as the best choice in terms of both average solution quality and fesibility for both data sets.

4.2 Performance of Local Search

In this part of the experiment, we assess the contribution of the local search in its rol of attempting to improve the quality of solutions and, more important, attempting to recover feasibility with respect to the solutions generated in the construction phase. To this end we run the procedure on 15 DU05 instances setting a budget limit of C = 20,000. The GRASP iteration limit was set at 1000.

Table 2. Improvement of local search.

	Construction			Local Search		
Instance	f	RCI	BCI	f	RCI	BCI
1	181.29	0	9.3	213.7	0	0
2	178.17	0	12.4	204.6	1.9	0
3	168.90	0	5.1	186.4	0	0
4	184.71	0	7.5	238.3	0	0
5	207.96	0	7.7	176.9	0	0
6	180.21	0	8.1	191.4	0	0
7	210.16	0	8.8	206.8	0	0
8	185.76	0	9.6	191.7	3.6	0
9	172.81	0	11.1	228.4	0	0
10	203.86	0	10.3	197.5	0	0
11	166.99	0	8.5	188.6	0	0
12	181.33	0	7.1	175.4	0	0
13	182.97	0	7.2	181.2	0	0
14	174.58	0	11.6	202.8	0	0
15	200.43	0	9.5	223.4	6.2	0

Table 2 displays the results, where columns 2-4 show the objective function value, RCI, and BCI obtained after the construction phase, respectively, and columns 5-7 show the same statistics at the end of the local search. It can be observed that the local search was very successfull on recovering feasibility (12 out of 15). In several instances the objective was also improved. This clearly indicates the excellent performance of the local search in reducing practically to zero the infeasibilities found at the end of the construction phase.

4.3 Assessment of Adaptive Memory

In order to evaluate the adaptive memory component, we compare our algorithm with (AM) and without (NAM) adaptive memory in both set of instances. In the following tables we are showing the relative improvement improvement (RI) of AM over NAM, computed as

$$\mathrm{RI} = 100 \times \frac{f_{NAM} - f_{AM}}{f_{NAM}}.$$

A negative (positive) value indicates a decrement (increment) in the objective.

Table 3 displays the results for DU05 instances. In this type of instances the results between the two startegies are very similar and it is not clear AM provides an advantage. We can see that there are three infeasible solutions under NAM and four infeasible solutions under AM. When comparing only the feasible solutions, we can see that in average there is an improvement of 0.29% of AM over NAM in solution quality, which is not too large.

Table 3. Evaluation of adaptive memory on DU05 instances.

	NAM			AM			
Instance	f	RCI	BCI	f	RCI	BCI	RI (%)
1	213.71	0	0	178.97	0	0	16.2
2	204.64	1.9	0	249.58	0	0	-21.9
3	186.45	0	0	197.74	0	0	-6.0
4	238.36	0	0	253.95	0	0	-6.5
5	176.98	0	0	191.93	0	0	-8.4
6	191.43	0	0	188.31	0	0	1.6
7	206.85	0	0	189.28	0	0	8.4
8	191.70	3.6	0	179.45	2.8	0	6.3
9	228.42	0	0	220.76	2.3	0	3.3
10	197.57	0	0	204.57	0	0	-3.5
11	188.61	0	0	215.12	1.6	0	-14.0
12	175.41	0	0	176.20	0	0	-0.4
13	181.23	0	0	196.56	0	0	-8.4
14	202.85	0	0	187.51	0	0	7.5
15	223.42	6.2	0	225.37	6.2	0	-0.8
Average		0.8	0		0.9	0	0.29

Results for DU10 are shown in Table 4. In this case, AM turns out to be successfull, particularly in terms of finding feasible solutions. We can see that, in terms of comparing the objective function value, NAM does slightly better than AM. However, AM was succesfull on recovering feasibility by finding all 15 out of 15 feasible solutions. Procedure NAM failed in this regard in 20% of the instances. It can be concluded that the use of AM can result in a valuable strategy towards better design in terms of feasibility.

Table 4. Evaluation of adaptive memory on DU10 instances.

	NAM			AM			
Instance	f	RCI	BCI	f	RCI	BCI	RI (%)
1	212.63	0	0	190.39	0	0	10.4
2	221.88	3.0	0	193.58	0	0	12.7
3	188.37	0	0	242.14	0	0	-28.5
4	242.58	0.6	0	200.62	0	0	17.2
5	182.15	0	0	192.21	0	0	-5.5
6	227.27	0	0	240.25	0	0	-5.7
7	177.41	0	0	188.52	0	0	-6.2
8	191.59	0	0	190.43	0	0	0.6
9	188.08	0	0	196.87	0	0	-4.6
10	185.89	0	0	193.90	0	0	-4.3
11	165.82	0	0	192.55	0	0	-16.1
12	190.82	0	0	196.49	0	0	-2.9
13	196.94	0	0	195.65	0	0	0.6
14	175.59	0	0	191.76	0	0	-9.2
15	169.06	0	0	171.81	0	0	-1.6
Average		0.24	0		0	0	-5.86

4.4 Assessment of Strategic Oscillation

As we recall from Section 3.2, in the proposed strategic oscillation parameters σ and γ penalize two terms in the merit function (5) changing dynamically whenever a movement is made and certain conditions are met. In this section we assess the performance of this strategy by comparing the algorithm with (SO) and without (NSO) the strategic oscillation. Both use the adaptive memory component. For NSO, we fixed parameters $\sigma = 10$ and $\gamma = 10$ and run the algorithm to observe the effect of not having a dynamic oscillation.

Tables 5 and 6 display the comparison for data sets DU10 and DU05, respectively. The information is similar to the one presented in the previous tables. In Table 5 we see that all the solutions reported by NSO satisfy the routing budget constraint, but not the balance constraints. The average relative infeasibilities are relatively high. This means this strategy struggled on trying to recover feasibility. Solutions obtained under SO are all feasible. The objective function values are better under SO as well. In Table 6 a similar behavior is observed. In this case, NSO could not obtain any feasible solution, and SO was successfull in finding feasible solutions in 11 out of 15 instances.

Table 5. Evaluation of strategic oscillation on DU10 instances.

Instance	NSO			SO		
	f	RCI	BCI	f	RCI	BCI
1	218.00	0	5.5	190.39	0	0
2	233.96	0	12.0	193.58	0	0
3	192.67	0	5.1	242.14	0	0
4	183.99	0	5.9	200.62	0	0
5	261.74	0	5.2	192.21	0	0
6	238.43	0	4.3	240.25	0	0
7	200.25	0	4.2	188.52	0	0
8	231.74	0	6.4	190.43	0	0
9	192.25	0	7.0	196.87	0	0
10	243.85	0	7.7	193.90	0	0
11	187.31	0	6.0	192.55	0	0
12	227.93	0	5.7	196.49	0	0
13	196.04	0	8.1	195.65	0	0
14	211.91	0	5.5	191.76	0	0
15	198.04	0	4.0	171.81	0	0

Table 6. Evaluation of strategic oscillation on DU05 instances.

Instance	NSO			SO		
	f	RCI	BCI	f	RCI	BCI
1	225.76	0	7.7	178.97	0	0
2	265.12	0	9.9	249.58	0	0
3	244.91	0	5.8	197.74	0	0
4	218.49	0	8.8	253.95	0	0
5	184.34	0	7.2	191.93	0	0
6	244.39	0	7.5	188.31	0	0
7	248.76	0	8.9	189.28	0	0
8	230.26	0	10.5	179.45	2.8	0
9	199.21	0	8.8	220.76	2.3	0
10	221.97	0	7.9	204.57	0	0
11	205.56	0	6.4	215.12	1.6	0
12	219.67	0	6.1	176.20	0	0
13	203.67	0	6.0	196.56	0	0
14	241.62	0	9.7	187.51	0	0
15	243.46	0	11.4	225.37	6.2	0

We can conclude directly that the strategic oscillation had a very positive impact leading the algorithm to feasible solutions and improving its solution quality. This clearly shows the excellent performance of the strategic oscillation.

5 Conclusions

We introduced a commercial Territory Design Problem with a routing budget constraint. This is, to the best of our knowledge, the first work to address both design and routing decisions within commercial territory design. To solve this problem we proposed a GRASP with some advance features such as adaptive memory and strategic oscillation.

The adaptive memory component was introduced during the construction phase as a diversification mechanism. The strategic oscillation was implemented within the local search to allow more flexibility in the search trajectory. Empirical evidence over two different classes of instances indicate the modest sucess of teh adaptive memori and the effectiveness of the strategic oscillation components. The incorporation of these two into the procedure helped not only improve the quality of the solutions but to recover feasibility for almost all of them. Adaptive memory helped in terms of finding feasible solutions particularly for the DU10 instances. For the DU05 instances, the use of adaptive memory did not provide a significant advantage. In contrast, it was observed that the use of strategic oscillation was very successfull on both obtaining feasible designs, and improving solution quality. Although these preliminary results are very promising, there is still room for improvement. For the more difficult instances (set DU05), for example, there were a few cases (4 out 15) where the use of SO was not enough to guarantee feasibility. The procedure can certainly be improved by means of other local search techniques.

Acknowledgments. This work was supported by the Mexican National Council for Science and Technology (CONACYT), grant SEP-CONACYT 48499-Y, Universidad Autónoma de Nuevo León under its Scientific and Technological Research Support, grants UANL-PAICYT CA1478–07 and CE012–09. The research of the first author has been funded by a fellowship for graduate studies from CONACYT.

References

1. Applegate, D., Bixby, R., Chvátal, V., Cook, W.: TSP cuts which do not conform to the template paradigm. In: Jünger, M., Naddef, D. (eds.) Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions, Lecture Notes in Computer Science, vol. 2241, pp. 261–304. Springer, Berlin, Germany (2001)
2. Bergey, P.K., Ragsdale, C.T., Hoskote, M.: A simulated annealing genetic algorithm for the electrical power districting problem. Annals of Operations Research 121(1), 33–55 (2003)

3. Bertolazzi, P., Bianco, L., Ricciardelli, S.: A method for determining the optimal disctricting in urban emergency services. Computers & Operations Research 4(1), 1–12 (1977)
4. Bozkaya, B., Erkut, E., Laporte, G.: A tabu search heuristic and adaptive memory procedure for political districting. European Journal of Operational Research 144(1), 12–26 (2003)
5. Caballero-Hernández, S.I., Ríos-Mercado, R.Z., López, F., Schaeffer, S.E.: Empirical evaluation of a metaheuristic for commercial territory design with joint assignment constraints. In: Fernández, J.E., Noriega, S., Mital, A., Butt, S.E., Fredericks, T.K. (eds.) Proceedings of the 12th Annual International Conference on Industrial Engineering Theory, Applications, and Practice. pp. 422–427. IJIE, Cincinnati, USA (november 2007)
6. Caro, F., Shirabe, T., Guignard, M., Weintraub, A.: School redistricting: Embedding GIS tools with integer programming. Journal of the Operational Research Society 55(8), 836–849 (2004)
7. Duque, J.C., Ramos, R., Suriñach, J.: Supervised regionalization methods: A survey. International Regional Science Review 30(3), 195–220 (2007)
8. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. Journal of Global Optimization 6(2), 109–133 (1995)
9. Fernández, E.A., Kalcsics, J., Nickel, S., Ríos-Mercado, R.Z.: A novel maximum dispersion territory design model arising in the implementation of the WEEE-directive. Journal of the Operational Research Society 61(3), 503–514 (2010)
10. Glover, F., Hao, J.K.: The case for strategic oscillation. Annals of Operations Research 183(1), 163–173 (2011)
11. Gutin, G., Punnen, A.P. (eds.): The Traveling Salesman Problem and its Variations. Springer, New York, USA (2002)
12. Haugland, D., Ho, S.C., Laporte, G.: Designing delivery district for the vehicle routing problem with stochastic demands. European Journal of Operational Research 180(3), 997–1010 (2005)
13. Hojati, M.: Optimal political districting. Computers & Operations Research 23(12), 1147–1161 (1996)
14. Kalcsics, J., Melo, T., Nickel, S., Gündra, H.: Planning sales territories: A facility location approach. In: Operations Research Proceedings 2001, pp. 141–148. Springer, Berlin, Germany (2002)
15. Kalcsics, J., Nickel, S., Schröder, M.: Towards a unified territory design approach: Applications, algorithms, and GIS integration. Top 13(1), 1–56 (2005)
16. Muyldermans, L., Cattryse, D., Van Oudheusden, D., Lotan, T.: Districting for salt spreading operations. European Journal of Operational Research 139(3), 521–532 (2002)
17. Pezzella, F., Nicoletti, B.: A system approach to the optimal health-care districting. European Journal of Operational Research 8(2), 139–146 (1981)
18. Ríos-Mercado, R.Z., Fernández, E.A.: A reactive GRASP for a commercial territory design problem with multiple balancing requirements. Computers & Operations Research 36(3), 755–776 (2009)
19. Salazar-Aguilar, M.A., Ríos-Mercado, R.Z., Cabrera-Ríos, M.: New models for commercial territory design. Networks and Spatial Economics (2011), (Forthcoming, doi: 10.1007/s11067-010-9151-6)
20. Zoltners, A.A., Sinha, P.: Sales territory alignment: A review and model. Management Science 29(11), 1237–1256 (1983)
21. Zoltners, A.A., Sinha, P.: Sales territory design: Thirty years of modeling and implementation. Marketing Science 24(3), 313–331 (2005)