A Tabu Search Approach for a Territory Design Problem with Stochastic Demands

Mónica Elizondo-Amaya[1], Roger Z. Ríos-Mercado[1], David P. Morton[2], and Erhan Kutanoglu[2]

[1] Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León
San Nicolás de los Garza, NL 66450, México
{moni,roger}@yalma.fime.uanl.mx
[2] Graduate Program in Operations Research & Industrial Engineering
The University of Texas at Austin
Austin, TX 78712, USA
{morton,erhan}@mail.utexas.edu

Abstract. Territory design can be defined as the problem of grouping small geographic units into larger geographic clusters called territories in such a way that the latter satisfy certain planning criteria. A stochastic version of a commercial territory design problem in presence of uncertain demands is addressed in this work. The objective is to minimize the territory imbalances with respect to the product demand subject to planning criteria such as territory connectivity, compactness, and balance with respect to the number of customers. A tabu search metaheuristic for large instances is developed for this problem. The proposed method incorporates advanced techniques such as strategic oscillation and specific neighborhood exploration strategies. Empirical evidence shows the value of using such advanced techniques. To the best of our knowledge, this is the first study addressing the stochastic nature of the demand in a commercial territory design problem.

Keywords: Commercial territory design, metaheuristics, tabu search, stochastic programming.

1 Introduction

In the general context, the territory design problem (TDP) may be viewed as the problem of grouping small geographic basic units (BUs) into larger geographic clusters, called territories, in such a way that the territories are acceptable (or optimal) according to certain planning criteria. This problem belongs to the family of districting problems that have a broad range of applications such as political districting (Hojati [1], Ricca and Simeone [2], Mehrotra et al. [3], Bozkaya et al. [4]) and sales territory design (Zoltners and Sinha [5], Fleischmann and Paraschis [6], Drexl and Haase [7]) to name the most relevant. The reader can find an extensive survey on approaches to districting problems in the works of

Kalcsics et al. [8] and Duque et al. [9]. The problem addressed in this paper is a commercial territory design problem motivated by a real-world application from the bottled beverage distribution industry. Given a set of city blocks, where two different activities are present in each block (number of customers and product demand)the firm wants to partition the area of the city into disjoint territories according to several criteria such as:(i) Balanced territories: Territories must be similar in size, with respect to each of the two block activity measures; (ii) connectivity: For each formed territory, BUs can reach each other by traveling within the territory; (iii) compactness: BUs assigned to a territory are relatively close to each other and (iv) a fixed number of territories.

Several variations of the deterministic version of this problem (that is, when the demands are known) have been studied in the past; however, the stochastic version of the problem has not been addressed before to the best of our knowledge. This work introduces the stochastic version of the commercial territory design problem (STDP) that considers the uncertainty of the customers demand. A solution approach based on tabu search (Glover [10]) is proposed to obtain solutions for this problem. The proposed algorithm incorporates advanced mechanisms to improve the local search performance such as strategic oscillation, a dynamic neighborhood generation technique and a candidate list strategy. Our empirical work includes an evaluation of the algorithmic performance of the proposed approach. Computational experiments report that the quality of feasible solutions is improved when these strategies are employed in the local search phase.

2 Problem Statement

Let $G = (V, E)$ be an undirected graph, 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 basic unit (BU) is associated with a node $j \in V$, and an edge connecting nodes i and j exists if BUs i and j are located in adjacent blocks. Multiple attributes are associated to each block $j \in V$ such as geographical coordinates (c_x^j, c_y^j), number of customers denoted by w_j and sales volume (product demand) which is a discrete random variable ξ_j with realizations indexed by $\omega \in \Omega$. The number of territories p is fixed and is given as a parameter. It is required that each node is assigned to only one territory. A p-partition of the set V is denoted by $X = (X_1, \ldots, X_p)$ where $X_k \subset V$ is called a territory of V. Let $w(X_k) = \sum_{i \in X_k} w_i$ denote the number of customers in the territory X_k. The balancing requirement with respect to the number customers is modeled by introducing a tolerance parameter τ. This tolerance parameter is user specified and it represents a limit on the maximum deviation from an ideal target allowed. This target value is given by the average size $\mu = \sum_{i \in V} w_i / p$. Another important feature is that all of the nodes assigned to each territory are connected by a path contained totally within the territory (i.e., each of the territories X_k must induce a connected subgraph of G). Figure 1 illustrates the balancing and connectivity constraints. In addition, it is required that in each of the territories, blocks

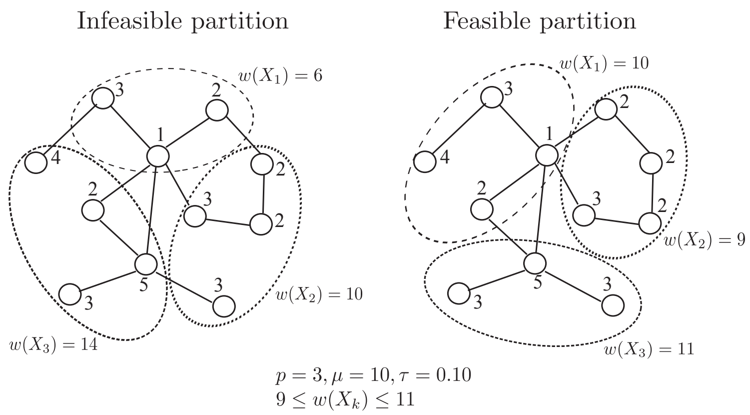

Fig. 1. Illustration of balancing and connectivity.

must be relatively close to each other. To account for this, we use a dispersion measure based on the p-center problem (pCP) and a maximal distance T for the dispersion of the territories is set. The dispersion of a territory X_k is given by $\max_{k=1,\ldots,p} \max_{j \in X_k} \left\{ d_{c(k),j} \right\}$, where $c(k)$ denotes the index of the center of the territory X_k, and is determined by:

$$c(k) = \arg \min_{j \in X_k} \min_{i \in X_k} \{d_{ij}\} .$$

Location of territory centers is not a requirement of the problem but a feature of the proposed formulation that was adopted for convenience when measuring dispersion of territories. Finally, the objective is to minimize the territory imbalances with respect to the stochastic demand. Let Π be the collection of all p-partitions of V and let $W(X_k) = \sum_{j \in X_k} \xi_j$ be the size of demand in the territory X_k. The combinatorial optimization model is given as follows:

Model (STDP)

$$\min_{X \in \Pi} \quad f(X) = \mathbb{E}\left[\max_{k \in K} \{W(X_k)\}\right] \qquad (1)$$

$$\text{subject to:} \quad \frac{w(X_k)}{\mu} \in [(1-\tau),(1+\tau)] \qquad k \in K, \qquad (2)$$

$$\max_{k=1,\ldots,p} \max_{j \in X_k} \left\{ d_{c(k),j} \right\} \leq T, \qquad (3)$$

$$G_k = G\left(V_k, E(V_k)\right) \text{ is connected} \qquad k \in K. \qquad (4)$$

Objective (1) minimizes the territory imbalances with respect to the product demand by minimizing the expectation of the maximum territory size with respect to the product demand. Constraints (2) represent the territory balance with respect to the number of customers as it establishes that the size of each territory with respect to this attribute must lie within a range (measured by tolerance parameter τ) around its average size. Constraint (3) establishes a compactness measure as it limits the maximum distance between a territory center and its assigned BUs. This parameter can be a policy established by the firm. Therefore, the problem can be formally described as finding a p-partition of the graph G that meets balance, connectivity and compactness constraints and minimizes the territory imbalances with respect to stochastic customer demands. This problem has been modeled in terms of integer programming and empirically validated using the Branch and Bound (B&B) procedure implemented by the commercial solver CPLEX 11.2. Small instances of just 20 nodes, 2 territories and 3 demand scenarios were solved to optimality. Due to the fact that real-world instances are even greater than 500 nodes, the use of a heuristic procedure is proposed. A tabu search procedure was developed and tested on a collection of data instances obtaining promising results.

3 Tabu search approach to STDP

Tabu search (Glover and Laguna [11]) is a well-known metaheuristic that has been widely used for successfully solving many combinatorial optimization problems. Particularly, in political districting, TS has shown an efficient performance [4]. Tabu search guides a local search procedure to explore the solution space beyond local optimality. Starting from an initial solution, TS iteratively moves from the current solution to its best neighbor, even if this new solution is worse than the one available, until a pre-specified stopping criterion is met. In order to avoid cycling and becoming trapped in local optima, certain moves (or solution attributes) that lead to previously explored regions are forbidden (i.e.,classified as tabu), forming the short-term memory of TS. The tabu status of a move may be overriden making it an allowable move if an aspiration criterion is satisfied (if, for instance, the tabu move leads to a new best solution). The tabu tenure is the length of time during which a certain move is classified as tabu. It can be kept constant or varied dynamically throughout the search. These key aspects of the TS algorithm tailored to STDP in this paper are described below.

3.1 Initial solution generation

At a given iteration, the construction phase consists of building p territories simultaneously in such a way that connectivity is always satisfied while infeasibility in terms of balance and dispersion is allowed to some extent. This procedure starts by selecting p territory seed centers $c(1), \ldots, c(p)$ which are the first BUs assigned to each territory (i.e., $c(k) \in X_k, k \in K = \{1, \ldots, p\}$). Selecting seed centers by using a purely random approach could lead to obtain inappropriate initial territory centers (e.g., centers close to each other). To avoid this, the problem of choosing an appropriate set of p initials seeds is viewed as a p-Dispersion Problem (Erkut et al. [12]). Territories are then built iteratively in three stages. In the first stage, a fraction δ of the total of BUs are iteratively assigned to territories by considering the threshold distance T and connectivity. At each iteration, a BU $i \in V$ is assigned to an adjacent territory X_{k^*} (i.e., the basic unit

i is connected by an edge to a BU already assigned to X_{k^*}) such that:

$$k^* = \arg \min_{k \in K_T(i)} E\left[\max\left\{W(X_1), \ldots, W(X_k \cup \{i\}), \ldots, W(X_p)\right\}\right], \quad (5)$$

where $K_T(i) = \{k \in K | d_{c(k),i} \leq T\}$. The latter process is iterated until a fraction δ of the total of BUs have been assigned to p territories, where the centers $c(1), \ldots, c(p)$ are updated every L iterations. From this stage, the p territories have been built through a greedy function that completely ignores the customer balance constraints. The rationale behind this is that at the start of the construction phase all territories are likely to violate these constraints thus it is not suitable to penalize them until they have a considerable size with respect to this activity measure. The second stage of the construction phase tries to assign the remaining nodes that were not assigned in stage one. For this stage a BU is assigned to an adjacent territory by considering both the objective function and balance constraints. For each basic unit $i \in V$, the cost of assign it to a territory X_k, $k \in K_T(i)$ is evaluated according to the following merit function:

$$\begin{aligned}\phi_k(i) = \lambda \left(\frac{1}{\gamma}\right) E\left[\max\left\{W(X_1), \ldots, W(X_k \cup \{i\}), \ldots, W(X_p)\right\}\right] \\ + (1-\lambda)\left(\frac{1}{\mu}\right) \max\left\{w(X_k \cup \{i\}) - (1+\tau)\mu, 0\right\} \quad (6)\end{aligned}$$

where $\gamma = \sum_{i \in V} E(\xi_i)/p$ (i.e., the expected value for the average size of demand in the territories), is used for normalizing the objective function. The second term of the Equation (6) represents the represents the relative infeasibility with respect to the upper bound of the balance requirement. Both factors objective function and balancing infeasibility are weighted by a parameter λ. A candidate list (RCL) restricted by a quality parameter α is constructed, from which a territory is randomly selected to assign the current basic unit. The process is repeated for every basic unit i. If a territory exceeds the average weight for the number of customers it is considered closed and no further node can be assigned to it. The latter process iterates until no basic unit can be assigned to a territory within the threshold distance T, or every territory is considered closed or if all BUs have been assigned. Since previous stages do not guarantee that all nodes will be assigned to a territory, a final stage is applied in which unassigned nodes are assigned by using Equation (5) or to their nearest adjacent territory in case that no territory center is within the maximal dipersion distance. We now describe the mechanisms for improving the constructed initial solution.

3.2 Neighborhood generation

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 compactness constraints as much as possible. Insert moves and pairwise exchanges (swaps) are frequently used move types for neighborhood generation in combinatorial problems. Let $X_{t(i)}$ denotes the territory to which node i belongs, $t(i) \in 1, \ldots, p$ we next define two different moves for the neighborhoods construction:

- $move_A(i,k)$: An insertion move reassign a basic unit i from its current territory $X_{t(i)}$ to another territory X_k; $t(i) \neq X_k$; $t(i), k \in 1, \ldots, p$.
- $move_B(i,j)$: A swap move exchanges basic units i and j between its respective territories $X_{t(i)}$ $X_{t(i)}$ and $X_{t(j)}$; $t(i) \neq t(j)$; $t(i), t(j) \in 1, \ldots, p$.

Connectivity must be kept, then only moves $move_A(i,k)$ where $X_{t(i)} \setminus \{i\}$ remains connected and for any $l \in X_k$, $\exists (i,l) \in E$ are allowed. The same criteria apply for moves $move_B(i,j)$. We denote as $N_A(X)$ and $N_B(X)$ the neighborhoods of a solution X generated from $move_A(i,k)$ and $move_B(i,j)$ respectively.

3.3 Merit function and strategic oscillation

A variant of the objective function is used to guide the local search phase. This function weights the objective function value and the infeasibility with respect to the compactness and balancing constraints is used. Specifically, the merit function for a given territory design $X = (X_1, \ldots, X_p)$ is given by:

$$F(X) = \left(\frac{1}{\gamma}\right) E\left[g(X)\right] + \beta_1 f_1(X) + \beta_2 f_2(X), \tag{7}$$

where,

$$g(X) = \left(\frac{1}{\gamma}\right) \max_{k=1,\ldots,p} \{W(X_k)\} \tag{8}$$

$$f_1(X) = \left(\frac{1}{d_{\max}}\right) \max_{k=1,\ldots,p} \left\{ \max_{j \in X_k} \left\{ d_{c(k),j} \right\} - T, 0 \right\} \tag{9}$$

$$f_2(X) = \sum_{k=1}^{p} \left(\frac{1}{\mu}\right) \max\left\{ w(X_k) - (1+\tau)\mu, (1-\tau)\mu - w(X_k), 0 \right\} \tag{10}$$

Then, the quality of a move is determined by (7). Equation (8) is the normalized objective function. Expresion (9) represents the relative infeasibility with respect to the compactness requirement. Equation (10) is the sum of all relative infeasibilities of the balancing constraints. Finally, β_1 and β_2 are penalty parameters to be dynamically updated according to strategic oscillation (Glover [13]). The penalty coefficients are then self-adjusted each r iterations. If compactness constraint was violated in all the r^* previous solutions, then $\beta_1 = \psi\beta_1$, $\psi > 1$ and the same occurs with β_2 when the balance constraints are violated in all r^* previous solutions. If all r^* previous solutions were feasible, the values of both parameters are reduced as follows: $\beta_q = \frac{1}{\psi}\beta_q$, $q = 1, 2$, $\psi > 1$. Otherwise β_q remains unchanged. The factor ψ as well as the values for r and $r*$ are user-defined parameters. With this strategy we can guide the search to a larger space by allowing infeasible moves.

3.4 Neighborhood exploration strategy

The proposed TS uses a dynamic neighborhood exploration strategy. The neighborhood structure is not a static set as it can change according to an infeasibility threshold $\epsilon \in [0,1]$. Since insertion moves are expected to perform a better job than swap moves in terms of recovering feasibility with respect to balancing constraints, then if the sum of relative infeasibilities in a solution X is greater than a threshold ϵ (i.e., $f_1(X) + f_2(X) > \epsilon$), the neighborhood $N_A(X)$ is constructed, otherwise the neighborhood structure $N_A(X) \cup N_B(X)$ is then considered.

3.5 Recency-based memory and tabu tenure

Whenever a $move_A(i,k)$ or $move_B(i,j)$ is performed, any move that puts the BU i back into territory $t(i)$ or BU j back into territory $t(j)$ is declared tabu and forbidden for the next θ iterations. In this study θ is randomly selected in some interval $[\theta_{\min}; \theta_{\max}]$ following a uniform distribution, where $\theta_{\min}$ and $\theta_{\max}$ are user-defined parameters. Then, a new tabu tenure is generated for the attributes that becomes tabu at a given iteration. This is a well known strategy in TS literature to induce a good balance between intensification and diversification (see [14]). The tabu status of a move is overridden making it allowable if such move leads to a new best solution.

3.6 Candidate list strategy

Considering every possible move from the current solution to select the best neighbor may be extremely time consuming and computationally expensive. A list of candidates is proposed were only those moves involving the k_1 worst territories with respect to quality (i.e., those with larger demand) and the worst k_2 with respect to infeasibility (i.e., those with the highest values $f_1 + f_2$) are considered. The aim of this selection is to avoid performing moves that do not reflect a significant improvement in the evaluation of the merit function.

4 Computational evaluation

In this section we present the preliminary experimental results on the strategic oscillation and the neigborhood exploration mechanism in our local search algorithm. The proposed TS was coded in C++ and all experiments were carried out on a Intel Core i5, 2.30 GHz computer. For the experiments, we generated randomly problem instances based on real-world data provided by the industrial partner. This data set is generated according to the characteristics described in [15]. In that work, full details on how the instances are generated can be found. We experimented with 10 instances of each size $(n,p) \in \{(100,6),(500,10)\}$. For all instances, the allowable deviation for the balancing constraints was set at 5% ($\tau = 0.05$). For this preliminar study, the random demand is described by 10 discrete scenarios, each with a fixed probability. The scenario probabilities are: 0.01, 0.04, 0.15, 0.02, 0.34, 0.14, 0.09, 0.1, 0.06, 0.05. The process terminates when $maxIter_1$ iterations have been performed or when $maxIter_2$ consecutive iterations have been performed without improving the best solution found during the search. The parameters of the algorithm were set as described in Table 1.

(n,p)	T	δ	α	λ	ψ	$\theta_{\min}$	$\theta_{\max}$	ϵ	r	r^*	k_1	k_2	$maxIter_1$	$maxIter_2$
(100,6)	200	0.5	0.4	0.7	2	5	10	0.003	10	3	2	2	1000	250
(500,10)	150	0.5	0.4	0.7	2	5	10	0.003	10	3	4	4	1000	250

Table 1. Algorithm parameters.

4.1 Strategic oscillation efficiency

The improvement produced when the strategic oscillation is incorporated into the local search is first addressed. To this aim, both penalty coefficients β_1 and β_2 in the merit function (Equation (7)) are fixed to a value φ (1,10,100) and solutions obtained (NSO solutions) are compared with those generated when the penalty coefficients are self-autoadjusted (SO solutions).

Instance	SO	NSO			RD(%)		
		$\varphi=1$	$\varphi=10$	$\varphi=100$	$\varphi=1$	$\varphi=10$	$\varphi=100$
1	24.3234	-	25.2348	24.9342	-	3.7470	2.5112
2	22.0237	-	23.149	25.2034	-	5.1095	14.4376
3	24.9014	-	26.7921	28.2347	-	7.5927	13.3860
4	25.6236	26.8923	29.1356	26.1073	4.9512	13.7061	1.8877
5	23.1421	-	24.2997	24.7231	-	5.0021	6.8317
6	24.3513	-	24.9762	27.1892	-	2.5662	11.6540
7	27.6923	-	27.6923	27.9217	-	0.0000	0.8284
8	20.2892	-	22.2349	22.8592	-	9.5898	12.6668
9	24.168	27.02334	23.9152	23.7462	11.8145	-1.0460	-1.7453
10	24.2107	-	25.8551	24.5266	-	6.7920	1.3048

Table 2. Performance of strategic oscillation for 100-node instances.

Instance	SO	NSO		RD(%)	
		$\varphi=10$	$\varphi=100$	$\varphi=10$	$\varphi=100$
1	17.4282	19.3927	20.2254	11.2720	16.0499
2	15.3521	17.2589	17.8924	12.4205	16.5469
3	18.9346	23.0282	24.4627	21.6197	29.1958
4	17.5503	20.4339	19.2389	16.4305	9.6215
5	19.8923	21.9934	23.4602	10.5624	17.9361
6	18.0034	21.3467	21.5469	18.5704	19.6824
7	17.3943	19.3992	20.0429	11.5262	15.2268
8	17.7723	20.389	19.4533	14.7235	9.4585
9	16.5814	19.5496	19.2369	17.9008	16.0149
10	20.4502	24.2421	25.9085	18.5421	26.6907

Table 3. Performance of strategic oscillation for 500-node instances.

Results of the empirical comparison for 100 nodes are summarized in Table 2. Columns 2 to 5 show the objective function value of solutions obtained from SO and NSO strategies. A relative deviation (RD) between SO and NSO solutions is computed as $100 \times [(sol(\text{NSO}) - sol(\text{SO}))/sol(\text{SO})]$. As it can be noticed, in 90% of the instances the quality of the SO solutions is better than NSO solutions for $\varphi = 10$ and 100 with average relative improvement of 5.30% and 6.37% respectively. For the smallest penalty coefficient value ($\varphi = 1$) no feasible solutions were identified in 80% of the instances. This superiority in the quality of the solutions generated by using strategic oscillation is better apreciated in

Table 3 for 500-node instances where the relative average RD are 15.35% and 17.64% for $\varphi = 10$ and $\varphi = 100$ respectively.

4.2 Evaluation of the neighborhood exploration strategy

The efficiency of using a dynamic neighborhood structure that variates depending on the level of relative infeasibility at each iteration is tested. As in the previous experiment, solutions obtained by using a dynamic neighborhood exploration strategy (DN) are compared with those obtained when the proposed TS procedure uses an static neighborhood (SN). Then, neighborhood $N_A(X)$ is considered for the first 500 iterations, after that the neighborhood $N_A(X) \cup N_A(X)$ is used. Computational results presented in Table 4 confirm the solution quality improvement when the neighborhood structure is chosen by using a threshold ϵ for the relative infeasibility. The relative deviations (computed as in the previous test) between both DN and SN solutions are on average 1.76% and 15.32% for 100 and 500 node instances, respectively, which is a remarkably high improvement particulary in the case of 500-node instances.

Instance	(n,p)=(100,6)			(n,p)=(500,10)		
	SN	DN	RD(%)	SN	DN	RD(%)
1	24.3234	24.7821	-1.8509	19.4356	17.4282	11.5181
2	22.0237	20.9237	5.2572	17.1267	15.3521	11.5593
3	24.9014	25.6221	-2.8128	20.1514	18.9346	6.4263
4	25.6236	25.6082	0.0601	20.6567	17.5503	17.7
5	23.1421	22.0732	4.8425	23.0561	19.8923	15.9046
6	24.3513	23.1892	5.0114	22.3783	18.0034	24.3004
7	27.6923	26.6351	3.9692	22.7923	17.3943	31.0332
8	21.6371	22.3146	-3.0361	19.0342	17.7723	7.1004
9	24.168	23.8375	1.3865	18.5168	16.5814	11.6721
10	24.2107	23.1003	4.8069	23.7197	20.4502	15.9876

Table 4. Comparison between static and dynamic neighborhood exploration strategies.

5 Conclusions and work in progress

In this paper, we introduced the stochastic version of a commercial territory design problem. A tabu search algorithm is presented for this problem. To the best of our knowledge the demand uncertainty has not been treated before in commercial territory design. The proposed TS procedure incoporates sophisticated mechanisms such as strategic oscillation, which constitute the core of many adaptive memory programming algorithms, and strategies for neighborhood exploration. Preliminary experimental results show the positive impact of these mechanisms as the quality of the solutions is significantly improved when these mechanisms are incorporated into the local search phase.

Several issues are still work in progress. Currently we are working on the calibration of parameters, we are studying the self-tuning of the algorithm parameters during the search according to evolution status. We plan to execute a more extensive experimentation considering instances of 1,000 BUs and up as well as increasing the number of demand scenarios. The study of the behavior of the other components of the local search such as the tabu tenure which can be adjusted during the search according to the search history is to be done. The incorporation of long-term memory strategies to guide the search into unexplored regions of the solution space (diversification) and performing a more thorough examination in some good or promising regions (intensification) may be needed as well.

Acknowledgments: This research has been supported by the Mexican National Council for Science and Technology (grant CB-2011-01-166397). In addition, the first author is supported by a fellowship for graduate studies from CONACYT.

References

1. Hojati, M.: Optimal political districting. Computers & Operations Research **23**(12) (1996) 1147–1161
2. Ricca, F., Simeone, B.: Local search algorithms for political districting. European Journal of Operational Research **189**(3) (2008) 1409–1426
3. Mehrotra, A., Johnson, E.L., Nemhauser, G.L.: An optimization based heuristic for political districting. Management Science **44**(8) (1998) 1100–1113
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) (2003) 12–26
5. Zoltners, A.A., Sinha, P.: Sales territory design: Thirty years of modeling and implementation. Marketing Science **24**(3) (2005) 313–332
6. Fleischmann, B., Paraschis, J.N.: Solving a large scale districting problem: A case report. Computers & Operations Research **15**(6) (1988) 521–533
7. Drexl, A., Haase, K.: Fast approximation methods for sales force deployment. Management Science **45**(10) (1999) 1307–1323
8. Kalcsics, J., Nickel, S., Schröder, M.: Toward a unified territorial design approach: Applications, algorithms, and GIS integration. TOP **13**(1) (2005) 1–56
9. Duque, J.C., Ramos, R., Suriach, J.: Supervised regionalization methods: A survey. International Regional Science Review **30**(3) (2007) 195–220
10. Glover, F.: Tabu search – Part I. ORSA Journal on Computing **1**(3) (1989) 190–206
11. Glover, F., Laguna, M.: Tabu Search. Kluwer, Boston, EUA (1997)
12. Erkut, E., Ülküsal, Y., Yeniçerioğlu, O.: A comparison of p-dispersion heuristics. Computers and Operations Research **21**(10) (1994) 1103–1113
13. Glover, F., Hao, J.K.: The case for strategic oscillation. Annals of Operations Research **183**(1) (2011) 163–173
14. Glover, F., Laguna, M., Martí, R.: Principles of tabu search. In T.González, ed.: Approximation Algorithms and Metaheuristics. Chapman & Hall/CRC (2007) 1–12
15. Ríos-Mercado, R.Z., Fernández, E.: A reactive GRASP for a commercial territory design problem with multiple balancing requirements. Computers & Operations Research **36**(3) (2009) 755–776