A GRASP for a Sales Territory Design Problem with Multiple Balancing Planning Criteria

Roger Z. Ríos-Mercado* **Elena Fernández†**

Abstract

In this paper we present a GRASP approach to a sales territory design problem with multiple node balance requirements. The problem, motivated by a real-world application in the soft drink industry, includes other planning criteria such as balancing among territories, and district connectivity. The objective is to find a feasible partition that minimizes a measure of territory compactness. A preliminary computational evaluation is included.

Keywords: combinatorial optimization, territory design, GRASP

1 Introduction

The sales territory design problem may be viewed as the problem of grouping small geographic sales coverage units (SCUs) into larger geographic clusters called sales territories in a way that the sales territories are acceptable (or optimal) according to relevant planning criteria. In particular, the problem addressed in this paper is motivated by a real-world application from a soft drink distribution firm. Although several sales territory design approaches have appeared in the literature [1, 2, 5], the specific features present in this concrete problem make it very unique, and not addressed before to the best of our knowledge. See Kalcsics et al. [4] for an extensive survey on approaches to sales territory design. Planning criteria include balancing, or designing territories that are similar in size with respect to each activity measure, connectivity, so customers can be visited within each territory, and territory compactness, so customers within a territory are relatively close to each other. In this work, we present and discuss a modeling framework for this

*Corresponding author, Graduate Program in Systems Engineering, Universidad Autónoma de Nuevo León, México, e-mail: *roger@mail.uanl.mx*

†Department of Statistics and Operations Research, Universitat Politècnica de Catalunya, Spain, e-mail: *e.fernandez@upc.es*

NP-complete combinatorial optimization problem. In addition, we propose and evaluate a GRASP approach to construct high-quality solutions. This includes a preliminary computational evaluation, which shows the behavior of our approach when compared to current industry practices.

2 Problem Description

The problem is modeled by a graph $G = (V, E)$, where a customer or SCU v is associated with a node and an arc is present between nodes i and j if customers i and j are located in adjacent blocks. Now each node $i \in V$ has certain parameters associated such as geographical coordinates (c_i^x, c_i^y), and three measurable activities. Let w_v^a be the value of activity a at node v, where $a = 1$ (number of customers), $a = 2$ (sales volume), and $a = 3$ (workload). The number of territories is given by the parameter p. It is required that each node is assigned to only one territory. Thus, the territories define a partition of V. One of the properties sought in a solution is that the territories are balanced with respect to each of the activity measures. So, let us define the size of territory B_k with respect to activity a as: $w^a(B_k) = \sum_{v \in B_k} w_v^a$. 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 activity measure. To account for this, we measure the balance degree by computing the relative deviation of each territory from its average size μ^a, given by $\mu^a = w^a(V)/p$. Another important feature is that all of the customers assigned to each territory are connected by a path contained totally within the territory. In other words, each of the territories B_k must induce a connected subgraph of G. In addition, industry demands that in each of the territories, customers must be relatively close to each other. One way to achieve this is to define a distance measure such as $D = \sum_{k=1}^{p} \max_{j \in B_k} d_{kj}$, where d_{kj} denotes the Euclidean distance from node j to center of territory k. All parameters are assumed to be known with certainty. The objective of the problem can be describe as finding a p-partition of V that satisfied the specified planning criteria such as balancing, contiguity, and compactness.

3 MILP Formulation

Indices and sets: Let n be the number of customers, a be an activity index ($a \in A = \{1, 2, 3\}$), k be a territory index ($k \in K = \{1, 2, \ldots, p\}$), and i, j be customer indices ($i, j \in V = \{1, 2, \ldots, n\}$).

Parameters: Let w_i^a be the value of activity a in node i ($i \in V$, $a \in A$), $w^a(B)$ the size of set B with respect to a ($a \in A$), p the number of territories, d_{ij} the Euclidean distance between i and j ($i, j \in V$), and μ^a the average value of a ($a \in A$).

Decision variables: Even though in the original problem we are not concerned with territory centers, we use these for modeling the compactness measure. So let $x_{ij} \in \{0,1\}$ be equal to 1 if unit j is assigned to territory with center in i, 0 otherwise ($i, j \in V$); and $y_j \in \{0,1\}$ equal to 1 if a territory center is placed at j, 0 otherwise ($j \in V$).

Model:

$$\begin{array}{llclll}
\text{Minimize} & f(x,y) & = & \sum_{i \in V} \max_{j \in V} \{d_{ij} x_{ij}\} & & (1) \\
\text{subject to} & \sum_{i \in V} x_{ij} & = & 1 & j \in V & (2) \\
& \sum_{j \in V} y_j & = & p & & (3) \\
& (1-\tau^a)\mu^a y_i & \leq & \sum_{j \in V} w_j^a x_{ij} \quad \leq \quad (1+\tau^a)\mu^a y_i & i \in V, a \in A & (4) \\
& & & G(B_k) \text{ must be connected} & k = 1, \ldots, p & (5) \\
& x_{ij}, y_i & \in & \{0,1\} & & (6)
\end{array}$$

Objective (1) measures territory compactness. Constraints (2) guarantee each node j is assigned to a territory. Constraint (3) sets the number of territories. Constraints (4) represent the territory balance with respect to each activity measure as it establishes that the size of each territory must lie within a range (measured by tolerance parameter τ^a) around its average size. It also assures that if no center is placed at i, no customer can be assigned to it. Note that there is an exponential number of constraints (5).

4 Solving the TDP by GRASP

GRASP [3], a fairly well-known metaheuristic that captures good features of both pure greedy algorithms and random construction procedures, has been widely used for successfully solving many combinatorial optimization problems. This type of approach has not been considered before for general territory design problems as far as we know. See Kalcsics et al. [4] for an extensive survey on approaches to the TDP.

A GRASP is an iterative process in which each major iteration consists of two phases: construction and post-processing. The construction phase attempts to builds a feasible solution S, and the post-processing phase attempts to improve it. When a feasible solution is successfully found in phase one, phase two is typically a local search around suitable neighborhoods with the aim of improving the objective function value. In our case, phase one does not necessarily terminates with a feasible solution, so the post-processing phase attempts to reduce the total relative infeasibility. Figure 1 illustrates a generic GRASP implementation in pseudocode. The algorithm takes as an input an instance of the TDP, the maximum number of GRASP iterations, and the restricted candidate list (RCL) quality parameter α, and returns a solution S.

`GRASP(P, limit_iterations, alpha)`
Input: A TDP instance P, GRASP iteration limit, and RCL quality parameter.
Output: A feasible assigment S.

$S^{best} \leftarrow \emptyset$;
for $l = 1, \ldots,$ `limit_iterations` do
    $S \leftarrow$ `ConstructGreedyRandomizedSolution(alpha)`;
    $S \leftarrow$ `PostProcessing(S)`;
    if (S better than S^{best}) then $S^{best} \leftarrow S$;
end for;
return S^{best};
stop;

Figure 1: A generic GRASP pseudocode.

The motivation for GRASP in this particular application stems from the fact that current state-of-the-art approaches based on two-stage location-allocation algorithms have hard time handling connectivity constraints (5) so they are better suited for problems where this type of constraints are not present. By handling these constraints within a construction heuristic such as GRASP one expects a better algorithmic behavior. In the construction phase, at a given iteration we consider a partial territory and attempt to either allocate an unassigned node to it or to "close" the current territory and start a new one. When assigning a node, a greedy function that weighs both a distance-based compactness measure and the relative violation of the balance constraints (4) is used.

Procedure `ConstructGreedyRandomizedSolution()` does not necessarily returns a feasible solution. In particular, constraints (3) and (4) may not be satisfied. To address this issue, a two-step post-processing phase is applied. First, if (3) is not met, i.e., the number of territories q found in phase 1 is different from p, we either merge territories ($q > p$) or split territories ($q < p$). The merging operation consists of iteratively considering a territory of smallest size and merge it with its smallest neighboring territory. This reduces the number of connected territories by one at each iteration. This is iteratively followed until $q = p$. The splitting operation consists of taking a territory of largest size, and split it into two connected territories (by recursively applying the same algorithm with $p = 2$). This increases the number of territories by one at each iteration, so the procedure is performed iteratively until $q = p$. Note that the merging operation can be done very efficiently, while the splitting operation is itself another TDP problem. However, the nature of the construction phase makes merging more likely to be applied than splitting.

After this first post-processing step, a second post-processing step consisting of a local search is performed. In this local search, a merit function that weighs both infeasibility with respect to (4) and objective function value is used. This second post-processing step is still under development, so the computational results will show an evaluation of our proposed approach with both the construction phase and the first post-processing phase.

5 Empirical Work

The proposed procedure was written in C++ and compiled with the Sun C++ compiler workshop 8.0 under the Solaris 9 operating system and run on a SunFire V440.

In this part of the work, our aim is to evaluate the GRASP algorithm with respect to the quality parameter α. For this purpose, we generated 10 random instances with data provided by a local firm as follows. First each instance topology was randomly generated as a planar graph in the $[0, 100] \times [0, 100]$ plane. Then, the three node activities were generated from a uniform distribution in the [4,24], [15,414] and [15,104] ranges for number of customers, product demand, and workload, respectively.

Table 1: Evaluation of GRASP parameter α for best D^{max}

	GRASP parameter α									
	0.1		0.3		0.5		0.7		0.9	
Instance	D^{max}	RSI	D^{max}	RSI	D^{max}	RSI	D^{max}	RSI	D^{max}	RSI
data500-01	362.1	3.6	354.1	4.0	363.0	4.2	376.7	4.0	392.1	6.4
data500-02	353.8	3.9	360.0	3.9	374.2	6.2	380.7	4.8	389.0	5.2
data500-03	354.8	4.0	359.9	3.1	370.6	4.1	384.3	3.1	391.7	3.9
data500-04	369.2	3.6	376.7	3.7	383.6	3.4	381.6	6.0	395.7	4.3
data500-05	360.5	2.8	361.0	3.5	368.8	3.8	384.4	3.9	389.7	5.5
data500-06	350.8	3.8	356.8	4.1	370.1	3.5	367.6	5.1	382.8	7.3
data500-07	361.2	3.9	362.4	4.2	370.8	4.9	376.7	5.5	392.1	3.4
data500-08	370.3	4.0	369.8	3.9	376.4	4.5	384.9	3.2	391.4	5.2
data500-09	349.3	3.7	360.1	4.1	367.4	4.1	373.1	4.3	379.4	6.2
data500-10	354.7	1.6	357.6	2.7	368.7	4.1	381.2	4.7	390.3	4.3

Table 1 shows a summary of the preliminary results when GRASP was applied to ten 500-customer instances. In each of these we use a target deviation parameter τ^a of 0.3 (i.e., a deviation of 30% from target). For these runs we set the GRASP iteration limit at 1000. Each row shows the results for a specific instance in terms of $D^{max} = \sum_k \max_{i,j \in B_k}\{d_{ij}\}$, and SRI is the sum of relative infeasibilities with respect to constraints (4) for the solution that attained the best value of D^{max} for several values of α. As can be seen from the table the best results in terms of compactness (distance measure D^{max}) have been obtained with small values of α. For eight out of the ten instances the best solution was obtained with $\alpha = 0.1$, whereas for the other two, the best solution was obtained with $\alpha = 0.3$. As was expected, the best solutions with respect to D^{max} do not necessarily correspond with the best solutions with respect to RSI.

Table 2: Evaluation of GRASP parameter α for best RSI

	GRASP parameter α									
	0.1		0.3		0.5		0.7		0.9	
Instance	D^{max}	RSI	D^{max}	RSI	D^{max}	RSI	D^{max}	RSI	D^{max}	RSI
data500-01	418.5	1.1	379.5	1.3	430.6	1.8	469.7	1.4	441.6	1.7
data500-02	414.9	1.9	440.5	1.7	421.8	1.9	447.0	2.5	446.8	2.3
data500-03	396.3	0.1	413.6	1.0	455.8	1.3	446.4	1.5	505.3	2.0
data500-04	405.0	0.1	423.4	1.2	445.2	1.4	452.8	1.7	501.4	1.9
data500-05	403.9	0.0	460.8	1.4	457.9	1.9	455.5	1.5	449.7	1.9
data500-06	403.1	0.0	439.4	1.9	460.5	1.9	422.2	2.2	431.9	2.3
data500-07	387.5	0.0	413.6	1.1	420.8	1.5	423.5	2.0	492.3	2.0
data500-08	426.4	0.0	423.3	0.7	427.7	1.8	481.6	2.1	482.3	2.1
data500-09	380.5	0.0	439.4	0.4	440.2	2.0	449.2	2.0	453.7	1.9
data500-10	401.4	0.0	421.2	0.4	443.9	2.0	433.5	2.1	432.3	1.4

Now we also keep track of the solutions with the lowest SRI values. Table 2 displays similar results, but this time showing the instance that delivered the lowest SRI value and its corresponding D^{max}. Again, the best results where obtained with small values of α. In particular, now for nine instances the best solution in terms of infeasibility was obtained with $\alpha = 0.1$ and for one instance the best solution was found with $\alpha = 0.3$.

In our experimentation we have observed that 20% of the solutions obtained with GRASP improved the solutions given by the local firm with respect to the two criteria (D^{max} and RSI). This fact indicates that the proposed model provides with a good trade-off for obtaining good quality solutions for all the relevant criteria for the problem.

Regarding computational effort, we observe that as we increase α CPU time goes up. For instance, running GRASP with $\alpha = 0.1$ takes about 0.06 s/iteration, while using $\alpha = 0.9$ takes about 0.09 s/iteration. This is expected since a larger value of α implies a larger RCL and therefore a longer storing and searching time. Another observation is that in 393 out of 50,000 phase 1 function calls, a solution with $q < p$ is found, meaning that in only 0.7 % of the cases a merging post-processing operation was performed.

6 Conclusions

In this paper we have presented a GRASP approach to a sales territory design problem with multiple node balance requirements. The problem that is motivated by a real-world application in the soft drink industry, includes several planning criteria such as balancing among territories, contiguity, and connectivity. The preliminary computational experimentation that has been performed has given good quality results and indicates that the considered model provides with an appropriate trade-off for the various criteria considered.

We are in the process of implementing a local search scheme that will attempt both to improve the objective measure and reduce the imbalance infeasibility hopefully to zero. We are also in the process of documenting a formal comparison with solutions generated by current industry methods, but it is very clear from our preliminary experiments that the proposed approach delivers solutions significantly better than current standards. In fact, in a run of the GRASP algorithm with 1000 iterations, approximately 32% of the GRASP instances dominate (in both objective and infeasibility measure) the current solution.

Acknowledgments: The research of the first and second author is supported by grant SAB2004–0092 of the Spanish State Secretary for Universities and Research under its Visiting Scholar Program and by grant TIC2003–05982-C05/4 of the Inter-Ministerial Spanish Commission of Science and Technology, respectively.

References

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

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

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

[4] J. Kalcsics, S. Nickel, and M. Schröder. Toward a unified territorial design approach: Applications, algorithms, and GIS integration. *Top*, 13(1):1–74, 2005.

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