Computational Experience with GRASP for a Maximum Dispersion Territory Design Problem

Roger Z. Ríos-Mercado * Elena Fernández † Jörg Kalcsics ‡
Stefan Nickel ‡

* Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León, Mexico
Email: roger@mail.uanl.mx

† Department of Statistic and Operations Research
Universitat Politècnica de Catalunya, Spain
Email: e.fernandez@upc.es

‡ Chair of Operations Research and Logistics
Universität des Saarlandes, Germany
Email: {j.kalcsics, s.nickel}@orl.uni-saarland.de

1 Introduction

We present a heuristic approach to a territory design problem motivated by a real-world application arising in the recollection of waste electrical and electronic equipment (WEEE) in European countries. The problem may be stated as follows. Given a set of recollection points for waste electrical and electronic goods, a central agency must assign each recollection point to a corporation that would take charge of the recollection. In countries like Germany and Spain, for instance, the coordination and supervision of the collection is done by a central registry which also determines the market share of all companies that sell equipment in the respective country. According to the law, this assignment must meet several requirements. For white goods (i.e., dish-washers, fridges, etc.), the assignment of stations to corporations should be made such that the average amount of returned WEEE items is proportional to the market share of the corporation. Moreover, the points assigned to a corporation should be evenly dispersed all over the country to avoid a monopolistic concentration. Note that this criterion is exactly the opposite of the usual compactness criteria of classical territory design problems [2]. For a general overview on territory design models and an introduction into the topic the reader is referred to [2].

In this paper, we present an empirical evaluation of three GRASP heuristics for this NP-hard combinatorial optimization problem. Three different construction mechanisms and three different local search schemes are derived and evaluated over a range of randomly generated instances. The results are very good. The local search strategies improve considerably the solutions found in the construction phase particularly in terms of their infeasibility status. All three procedures reported feasible solutions.

2 Problem Description

Description: White goods are further subdivided into devices that have freezing capabilities and those that do not (product type 1 and 2, respectively). The average amount of WEEE of a basic area is assumed to be proportional to the number of households. Moreover, recycling (measured by means of suitable performance indicators), all basic areas are classified into three groups: good, mediocre, and bad. A good basic area is for example one, which has a relatively small geographic extent, contains few collection stations, and possesses a good infrastructure. The motivation for this classification is that the actual costs for the recollection and recycling should also be (more or less) proportional to the market shares of the corporations. As the market shares may differ for the two product types, it is allowed to split basic areas, i.e., for some basic areas the corporation that collects type 1 products of type may not be the same as the one that is responsible for the type 2 products. The task is then to assign basic areas fully or partially to corporations such that (i) for both product types all basic areas are assigned to a corporation; (ii) for each corporation, the total number of households of all basic areas assigned to the corporation is proportional to its market share for each of the two product types; (iii) the good, mediocre, and bad basic areas are evenly distributed among the corporations relative to their market shares; (iv) the number of split basic areas is not too large; and (v) for each corporation, all basic areas that are fully or partially assigned to the corporation are as dispersed as possible. The set of all basic areas assigned to a corporation for at least one of the two product types is called a territory.

Model: *Indices and sets:* Let V be the set of basic units (BU), with $|V| = n$, and C the set of corporations, with $|C| = m$. Let P be set of product types. In this work $|P| = 2$. Let Q be the set of quality indices, where $q \in Q = \{1, 2, 3\}$ stands for good (1), mediocre (2), and bad (3). Let V^q be the set of BUs of quality q, $q \in Q$. *Parameters:* Let w_i represent the number of households of BU i, $i \in V$. Let q_i be the quality of BU i, $i \in V$. Let MS_k^p denote the market share of corporation k for product p, $k \in C$, $p \in P$. Let $d_{ij}]$ represent the Euclidean distance between i and j, $i, j \in V$. Let τ and β denote the relative tolerance associated with number of households market share proportionality and the relative tolerance associated with fair quality distribution of households, respectively. Let S be an upper limit on the number of allowed split BUs. *Computed parameters:* Let $w(B)$ $(= \sum_{i \in B} w_i)$ be the size of set B with respect to node activity, $B \subset V$, with $w(V) = W$. Let $c^q(B)$ $(= |B \cap V^q|)$ be the cardinality of set B with respect to BU quality q, $q \in Q$, $B \subset V$. *Decision sets:* X_k^p is the set of BUs assigned to corporation k for product p, $k \in C$, $p \in P$; X_k is the set of BUs assigned to corporation k for at least one product $p \in P$, $k \in C$, that is, $i \in X_k \Leftrightarrow i \in X_k^p$ for some $p \in P$; X^s is the set of split BUs, that is, $i \in X^s \Leftrightarrow \exists k_1, k_2 \in C, k_1 \neq k_2$ such that $i \in X_{k_1}^1 \wedge i \in X_{k_2}^2$. Note that for a given value of p, $X^p = (X_1^p, \ldots, X_m^p)$ is a partition of V. Let X_k be the set of all basic units assigned to territory k for at least one product p, i.e., $X_k = \cup_{p \in P} X_k^p$. *Model:* The problem consists of finding $|P|$ m-partitions $X^p = (X_1^p, \ldots, X_m^p)$; $p \in P$ so as to

$$\text{Maximize} \qquad f(X) = \min_{k \in C} \min_{i,j \in X_k} \{d_{ij}\} \tag{1}$$

$$\text{subject to} \qquad \frac{1}{W} w(X_k^p) \in [(1-\tau)MS_k^p, (1+\tau)MS_k^p] \quad k \in C, p \in P \tag{2}$$

$$\frac{1}{|V^q|} c^q(X_k^p) \in [(1-\beta)MS_k^p, (1+\beta)MS_k^p] \quad q \in Q, k \in C, p \in P \tag{3}$$

$$|X^s| \leq S \tag{4}$$

Objective (1) measures territory dispersion. Constraints (2) represent the territory balance of the number of households with respect to the market share proportion for each product (measured by tolerance parameter τ). Constraints (3) assure a fair distribution of the basic areas regarding its product quality (measured by tolerance parameter β). Constraint (4) sets a limit on the number of maximum split BUs allowed. We call this problem the *Maximu-Dispersion Territory Design Problem* (MaxD-TDP). MaxD-TDP is NP-hard since we can reduce the well-known Partition Problem to it.

3 Description of Heuristics

GRASP [1] is a well-known multi-start metaheuristic that captures good features of both pure greedy algorithms and random construction procedures. Ecah GRASP iteration consists of two phases: construction and local search. The construction phase builds a solution, and the local search phase attempts to imrpove it. The best overall solution is kept as the result. construction procedures. It has been successfully used for solving many combinatorial optimization problems, including territory design [3].

Construction Phase: Two of the procedures (1 and 1R) try to assign basic areas that are relatively close to each other to different corporations. The third one (Procedure 2) is based on the complementary idea of trying to assign to the same corporation basic areas that are relatively far away from each other.

Procedure 1: After sorting the pairwise distances in non-decreasing order, we go through this list step by step, starting with the smallest non-zero value. At a given iteration, the next largest distance d_{ij} is considered and if i and/or j are yet unassigned, we try to allocate them to different corporations. The allocation decision is hereby based on a greedy function that takes a distance-based measure and the sum of the relative violations of the upper balancing constraints into account. More precisely, if $X = (X_1, \ldots, X_m)$ is the partial solution obtained so far, for a given basic area i, a corporation k, and a product p, the relative violation of the upper bound (UB) of the market share balancing constraints (2) and the UB of the logistics quality balancing constraints (3) is given as

$$G_i^p(k) = \max\left\{\frac{w(X_k) + w_i}{W \cdot MS_k^p} - (1+\tau),\, 0\right\} + \max\left\{\frac{c^{q_i}(X_k) + 1}{|V_{q_i}| \cdot MS_k^p} - (1+\beta),\, 0\right\}$$

where q_i is the quality index of BA i. Note that it makes no sense to include the lower bound violations at this point, as they will always be violated until the very end of the construction procedure. The greedy function is then defined as $\phi_i(k) = \lambda\, F_i(k) - (1-\lambda)\, G_i(k)$, where $\lambda \in [0,1]$, $G_i(k) = G_i^1(k) + G_i^2(k)$ and $F_i(k) = d_i(X_k) := \min_{j \in X_k} d_{ij}$ (in case $X_k = \emptyset$, then $d_i(X_k) := \infty$). Figure 1 shows the pseudocode of

function `ConstructGreedyRandomized_1` (α)
Input: $\alpha :=$ GRASP RCL quality parameter.
Output: An assignment $X = \{X_k\}_{k \in C}$.

$X_k = \emptyset, k \in C$; $V^a = \emptyset$ (set of assigned basic areas);
sort $\{d_{ij}\}$, $i \neq j$, and store them in DL;
while ($|V^a| < |V|$ AND $|DL| > 0$) **do**
    $d_{ij} \leftarrow D^{(1)}$ (first element in DL); $DL \leftarrow DL \setminus \{D^{(1)}\}$;
    if ($i \in V^a$ AND $j \in V^a$) **go to** step 2;
    if ($i \notin V^a$ AND $j \in V^a$) [Note: opposite case is symmetric]
        compute greedy function $\phi_i(k)$, $k \in C$;
        build RCL; $\hat{k} \leftarrow$ rand(RCL);
        if ($\hat{k} =$ territory of j) **then** RCL $\leftarrow$ RCL$\setminus\{\hat{k}\}$; $\hat{k} \leftarrow$ rand(RCL); **endif**;
        assign i to corporation $\hat{k}$: $X_{\hat{k}} \leftarrow X_{\hat{k}} \cup \{i\}$; $V^a \leftarrow V^a \cup \{i\}$;
    endif;
    if ($i \notin V^a$ AND $j \notin V^a$)
        compute greedy function $\phi_i(k)$; $k \in C$
        build RCL; $\hat{k} \leftarrow$ rand(RCL);
        assign i to corporation $\hat{k}$: $X_{\hat{k}} \leftarrow X_{\hat{k}} \cup \{i\}$; $V^a \leftarrow V^a \cup \{i\}$;
        compute greedy function $\phi_j(k)$, $k \in C$;
        build RCL; $k' \leftarrow$ rand(RCL);
        if ($k' = \hat{k}$) **then** RCL $\leftarrow$ RCL $\setminus\{k'\}$; $k' \leftarrow$ rand(RCL); **endif**;
        assign j to company k': $X_{k'} \leftarrow X_{k'} \cup \{j\}$; $V^a \leftarrow V^a \cup \{j\}$;
    endif;
endwhile;
return $X = \{X_1, \ldots, X_m\}$;
end `ConstructGreedyRandomized_1`

Figure 1: Construction Procedure 1.

this procedure. DL denotes the list of all pairwise distances ordered by non-decreasing values, with ties arbitrarily broken, and $D^{(r)}$ denotes the r-th element of the list. The restricted candidate list is defined as RCL $= \{k : \phi_i(k) \geq \phi_i^{\max} - \alpha(\phi_i^{\max} - \phi_i^{\min})\}$, where $\phi_i^{\min} = \min_k \phi_i(k)$ and $\phi_i^{\max} = \max_k \phi_i(k)$.

Procedure 1R: As the UB of constraints (2) and (3) have been taken into account in Procedure 1 through the greedy function only, is is likely to obtain infeasible solutions. Therefore, in a variant of Construction Procedure 1, we try to obtain solutions with no or at least fewer upper bound violations. Construction Procedure 1R has two phases. In the first one, basic areas are assigned to corporations as in Construction Procedure 1, provided that they do not violate the UB of (2) and (2). This can be achieved easily through a modified restricted candidate list: $RCL = \{k : G_i(k) = 0 \land \phi_i(k) \geq \phi_i^{\max} - \alpha(\phi_i^{\max} - \phi_i^{\min})\}$. As some basic areas may remain unassigned at the termination of the first phase, these areas are assigned to corporations in a second phase in a different fashion. Throughout this phase, to reduce the violation of the UB of (2) and (3), we use as greedy function just $G_i(k)$. Splitting of basic areas is allowed in the second phase when no more than $|S|$ basic areas remain unassigned.

Procedure 2: Using a different rationale, we now try to assign to the same corporation basic areas that are relatively distant from each other. The procedure has two phases. In the first, the iterative procedure builds m territories, one at a time, using a farthest insertion greedy function that assigns to the territory X_k currently being built BAs relatively “far away” from X_k. Since, again, no violation of the UB balancing constraints is allowed, the RCL only contains unassigned BAs for which $G_i(k) = 0$. Note that $\phi^{max} = \max_i \phi(i)$ and $\phi^{min} = \min_i \phi(i)$. If some BAs remain unassigned at the termination of this phase, we proceed to a second phase that is exactly as in Construction Procedure 1R.

Local Search: In this phase we attempt to recover feasibility as well as to improve the objective function value. Solutions are now evaluated by means of a function that weighs both infeasibility with respect to the balancing constraints as well as the smallest pairwise distance among the territories. This function is similar to the greedy function $\phi_i(k)$ used in the construction phase, however, with the difference that now the sum of relative infeasibilities $G_i(k)$ takes into account the violation of the lower bounds of balancing constraints (2) and (3) as well. The types of exchange moves that we consider are the following: *Type A1:* For all products, reassign a basic area i that is currently assigned to some territory X_k of corporation k to the territory $X_{k'}$ of a different corporation $k' \neq k$. The size of the neighborhood is nm. *Type A2:* Reassign a basic area i just for product p from its current territory to a different corporation's territory. Splitting is allowed. The size of the neighborhood is $2nm$. *Type B:* Exchange the assignment of basic areas i and j currently allocated to different corporations for one or both products. The size of the neighborhood is $2n^2$.

4 Empirical Work and Conclusions

We generated problem instances using real-world data. Basic areas correspond to German zip-code areas with their respective number of households. The instances range from 100 up to 300 basic areas in steps of 50, and with four up to seven corporations. We generated five instances for each number of basic areas, except for the last, where we have just four instances. In combination with the four different numbers of corporations, this yields in total 96 instances. The tolerances for the demands and for the logistics indices are $\tau = 0.05$ and $\beta = 0.2$, respectively. The pairwise distances were computed as the Euclidean distance between the polygonal representations of the zip-code areas. In that way, neighboring basic areas have distance zero. The logistics indices were chosen randomly such that we have approximately the same number of good, mediocre, and bad basic areas. We set S to 20% of the value of n for each instance. Finally, the market shares of the companies are computed independently for the two products. A market share is drawn uniformly from the interval $[\frac{0.75}{m}, \frac{1.25}{m}]$. At the end, the market shares are normalized to obtain a total sum of 1. The heuristic procedures were coded in C++ and compiled with the Sun C++ compiler workshop 8.0 under the Solaris 9 operating system. They were run on a SunFire V440 with 8 GB of RAM and 4 UltraSparc III procesors at 1062 MHz.

Parameter Tuning: For fine-tuning the RCL parameter α and the weight parameter λ of the greedy function, we run the different versions of the GRASP with an iteration limit of 500 and no local search, and measure both the degree of violation of the balancing constraints and the value of the objective function. The set of values for λ is $\{0.5, 0.6, \dots, 1.0\}$, and for α we take $\{0, 0.2, \dots, 1.0\}$. Figure 2, displays results for Procedure 1 with λ and α fixed, respectively. In these figures, the left vertical axis measures the average (over all instances) relative infeasibility with respect to the upper and lower balancing constraints, whereas the right vertical axis measures the average (over all instances) percent gap between the objective function value of the obtained solution and the best solution found with all the values of the tested parameters. Similar figures for the other two construction procedures are omitted here for space reasons, but will be presented at the conference. For Procedure 1, a value of $\alpha = 0.2$ consistently finds the best compromise between infeasibility and deviation from maximum dispersion for each tested value of λ. Comparing the values of λ we observe that $\lambda = 0.5$ yields a good compromise. Due to their mechanism for building solutions, λ plays a minor role in Procedures 1R and 2. Thus, only α was evaluated. We observe that $\alpha = 0.2$ and $\alpha = 0.8$ yielded the best compromise between deviation from maximum dispersion and infeasibility for each of these procedures.

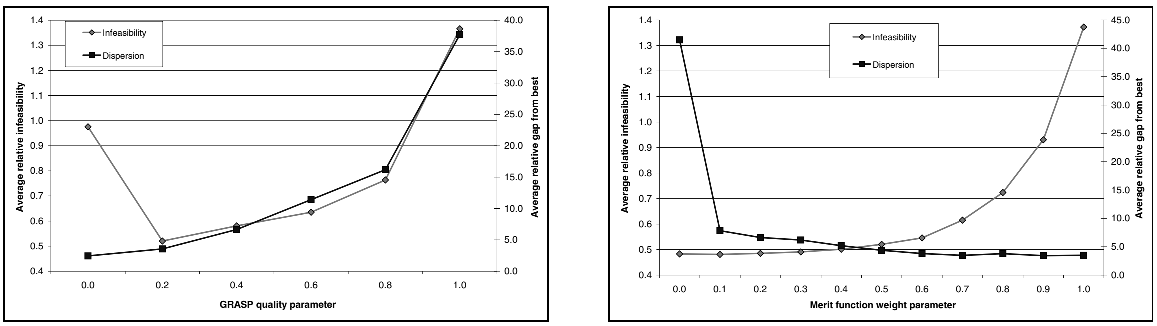

Figure 2: Construction Procedure 1 with $\lambda = 0.5$ as a function of GRASP quality parameter (left-hand side) and with $\alpha = 0.2$ as a function of the merit function weight parameter.

Local Search Strategies: Next, we investigate the behavior and effect of the local search. In the previous section we introduced three different neighborhoods. For our tests we consider three different combinations of these neighborhoods: LS1 (apply type A1 and then type A2 neighborhood), LS2 (apply type A2 and then type A1 neighborhood), and finally LS3 (apply type A2, then type A1, and then type B neighborhood). Since the neighborhoods are polynomially bounded and not too large, we use a best improvement strategy. For Constructive Procedure 1, 1R, and 2, we use (α, λ) values of $(0.2, 0.5)$, $(0.2, 1.0)$, and $(0.8, 1.0)$, respectively. The GRASP iteration limit was set to 2000. The results (omitted for space reasons) indicate that, independently of the construction procedure, both LS1 and LS2 were able to obtain always feasible solutions, which represents a significative improvement with respect to the construction phase where no feasible solutions were found (although solutions with relatively small deviations from feasibility were obtained). This shows the effectiveness of the local search strategies for repairing the infeasibility issue. The results also indicate that LS2 outperforms LS1 in terms of relative deviations from the best known value for the dispersion objective. In addition, LS2 uses considerably less CPU time than LS1 (about 40% less in average). Therefore, we conclude that it is more advantageous to explore A2 first, and then A1.

Heuristic Comparison: Table 1 shows a valid comparison among the heuristics under local search strategy LS2. This time the relative dispersion gap is computed with respect to best known value of the dispersion objective found by any procedure under any local search strategy. We observe that, in terms of solution quality, Construction Procedure 2 exhibits a very poor performance compared to the other two. Procedures 1 and 1R perform very similarly with respect to both dispersion objective and CPU time, with Procedure 1 being slightly better than 1R in terms of the dispersion objective. In particular, for the largest problems (300 basic areas), both heuristics find the same solution for 75% of the instances. For problem instances below 300 basic areas, Construction Procedure 1 finds the best solution. Overall, Procedures 1 and 1R each find 76 best solutions. It is important to highlight the tremendous benefit reported by the local search phase in each case. Most of the solutions found in the construction phase were infeasible (feasibility success of 5% and large deviations from feasibility in some cases). After the local search, feasibility was always recovered in all cases. Neighborhoods A1 and A2 helped to bring this figure up to 100.0%. This suggests further work on the local search schemes could be worthwhile. We also tested strategy LS3 for the 300-node instances with Procedure 1. For 11 out of 16 instances, no improvement was found. Moreover, while the overall average relative improvement is less than 1.5%, the average CPU time for LS3 is 19778.9 seconds, which represents, compared to the 1056.5 seconds for LS2, a very large increase that barely pays off in terms of solution quality.

n	Statistic	Proc 1	Proc 1R	Proc 2
200	Avg. rel. dispersion gap (% from best known)	1.5	0.2	24.1
	Worst relative dispersion gap (%)	13.3	3.6	44.6
	Number of best solutions	15	18	1
	Average CPU time	440	411	787
250	Avg. rel. dispersion gap (% from best known)	0.7	0.7	28.4
	Worst relative dispersion gap (%)	3.2	6.2	45.6
	Number of best solutions	14	15	0
	Average CPU time	699	671	1277
300	Avg. rel. dispersion gap (% from best known)	0.0	0.9	30.7
	Worst relative dispersion gap (%)	0.0	7.6	53.2
	Number of best solutions	16	12	0
	Average CPU time	1056	1067	2064
Total	Avg. rel. dispersion gap (% from best known)	0.7	0.9	24.6
	Worst relative dispersion gap (%)	13.3	16.6	53.2
	Number of infeasible solutions	0	0	0
	Number of best solutions	76	76	0

Table 1: Comparison of construction procedures under LS2. 100- and 150-BU instance results are now shown here but tallied in the total.

Conclusions: We have presented a computational study of a MaxD-TDP motivated by a real-world case in the recollection of WEEE. To the best of our knowledge, this is the first model based on maximum dispersion in the territory design literature. We have evaluated three construction procedures and three local search schemes within a GRASP framework over a range of instances randomly generated according to real-world scenarios This included heuristic fine tunning and a comparison of several local search strategies among heuristics with very good results. There are several challenging avenues for future research, for instance, the design of more sophisticated metaheuristics, and the study of an extension that would combine the design of the territories with the planning of the collection routes.

Acknowledgments: This research has been partially supported through grants MTM2006-14961-C05-01, of the Spanish Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica (I+D+I), SEP-CONACYT 48499-Y, of the Mexican National Council for Science and Technology, and PAICYT CA-1478-07, of the Universidad Autónoma de Nuevo León.

References

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

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

[3] 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.