Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 3690474, 13 pages
http://dx.doi.org/10.1155/2016/3690474

Research Article

A Comparison of Local Search Methods for the Multicriteria Police Districting Problem on Graph

F. Liberatore[1] and M. Camacho-Collados[2,3]

[1] *UC3M-BS Institute of Financial Big Data, Charles III University of Madrid, Getafe, 28903 Madrid, Spain*
[2] *Deputy Directorate for Operations, Spanish National Police Corps, 28010 Madrid, Spain*
[3] *Statistics and Operations Research Department, University of Granada, 18071 Granada, Spain*

Correspondence should be addressed to F. Liberatore; fliberat@inst.uc3m.es

Received 8 June 2015; Revised 29 January 2016; Accepted 16 February 2016

Academic Editor: Francisco Chicano

In the current economic climate, law enforcement agencies are facing resource shortages. The effective and efficient use of scarce resources is therefore of the utmost importance to provide a high standard public safety service. Optimization models specifically tailored to the necessity of police agencies can help to ameliorate their use. The Multicriteria Police Districting Problem (MC-PDP) on a graph concerns the definition of sound patrolling sectors in a police district. The objective of this problem is to partition a graph into convex and continuous subsets, while ensuring efficiency and workload balance among the subsets. The model was originally formulated in collaboration with the Spanish National Police Corps. We propose for its solution three local search algorithms: a Simple Hill Climbing, a Steepest Descent Hill Climbing, and a Tabu Search. To improve their diversification capabilities, all the algorithms implement a multistart procedure, initialized by randomized greedy solutions. The algorithms are empirically tested on a case study on the Central District of Madrid. Our experiments show that the solutions identified by the novel Tabu Search outperform the other algorithms. Finally, research guidelines for future developments on the MC-PDP are given.

1. Introduction

The Police Districting Problem concerns the definition of sound patrolling sectors in a police district. An extensive literature review on this family of problems is given by Camacho-Collados et al. [1]. The newest member of this family is the Multicriteria Police Districting Problem (MC-PDP) [1]. The novelty of this model stands in that it evaluates the workload associated with a specific patrol sector according to multiple criteria, such as area, crime risk, diameter, and isolation, and that it finds a balance between global efficiency and workload distribution among the agents, according to the preferences of a decision-maker (i.e., the service coordinator in charge of the patrolling operations in a police district). The MC-PDP was originally formulated in collaboration with the Spanish National Police Corps (SNPC) and it was solved by means of a fast heuristic algorithm that is capable of rapidly generating patrolling configurations that are more efficient than those adopted by the SNPC. When combined with Predictive Policing methodologies [2], the MC-PDP allows designing patrolling configurations that focus the distribution of resources on the most relevant locations, with a consequential improvement in the effectiveness of patrolling operations. This is the rationale of the paper by Camacho-Collados and Liberatore [3] that presented a Decision Support System (DSS) for the implementation of a paradigm of Predictive Patrolling in the SNPC.

The contributions of this paper are the following. In this research we extend the applicability and the quality of the solutions found by the MC-PDP with the objective of improving the performance of the DSS for Predictive Policing. In particular, we tackle one of the major limitations of the original formulation of the MC-PDP and propose and compare new heuristic algorithms. More specifically, the original MC-PDP was formulated to partition a grid. In this paper we formulate the MC-PDP to generate patrolling district on a generic graph, without any assumption on its topology. This allows for the definition of patrolling configurations using census districts as the atomic unit of patrolling. As explained by Sarac et al. [4] the use of a structure based on census districts is desirable as it allows easy access to demographic data and, at the same time, it is suitable for use by other agencies. Translating the MC-PDP to a generic graph requires the definition of an efficient and practical condition for set convexity that we derive from the classical definition of convexity in graphs. In terms of solution methodologies, we propose three local search algorithms for the MC-PDP on a graph, including a Tabu Search (TS). Thanks to its ability to escape from local optima and its versatility, the TS has been successfully applied to a very wide breadth of contexts and problems, such as parameter optimization [5], vehicle routing [6], hardware/software partitioning [7], and job shop scheduling [8]. The MC-PDP is a variant of the graph partitioning problem. The first application of the TS to the graph partitioning problem is due to Rolland et al. [9]. In recent years, the TS has been successfully applied to this family of problems, either individually or combined with other approaches [10–14].

The proposed algorithms are extensively tested on a real dataset based on a case study of the Central District of Madrid. Their performances are then compared and analyzed statistically. Finally, the best solutions found by the algorithms are illustrated and operational insights are drawn.

The remainder of the paper is structured as follows. The following section presents a review of the most relevant contributions to the literature. In Section 3 we formulate the MC-PDP for a generic graph and propose a methodology to deal with the problem of partitioning a generic graph into convex blocks. Next, we present in detail the local search algorithms developed for the solution of the models. In Section 5 we explain the dataset and the computational experiments run to test the algorithms. Also, we analyze the results and provide insights on the solutions obtained. Finally, we conclude the paper with some guidelines for future research.

2. Related Work

In his seminal work, Mitchell [15] proposes a clustering heuristic for the redesign of patrol beats in Anaheim, California. The underlying optimization model considers both the total expected weighted distance to incidents and a workload measure defined as the sum of the expected service and travel time. A different approach is presented by Bodily [16] that adopts a utility theory model that takes into consideration the preferences of the citizens, the administrators, and the service personnel. The problem is solved by means of a local search algorithm that explores the solution space by swapping elements between the sectors. Benveniste [17] includes for the first time workload equalization criteria. The final model was nonlinear and stochastic in nature and was solved by means of an approximation algorithm. D'Amico et al. [18] propose a simulated annealing algorithm. The solutions identified by the search algorithm are evaluated by an external software program, PCAM [19, 20], that calculates sector workloads. The external routine is based on a queuing model that computes relevant statistics regarding a sector, including the optimal number of cars to be allocated. Equity in terms of area to be patrolled is enforced in the model by constraining the ratio of the size of the largest and the smallest sectors. A simpler approach is presented by Curtin et al. [21] that tackle the problem of partitioning a police district by using a covering model that maximizes the number of incidents that are close to the centers of the sectors. Five years later, the authors extend their approach [22] and include backup coverage (e.g., multiple coverage of high priority locations). Finally, Zhang and Brown [23] propose a heuristic algorithm for the generation of districting, evaluated using an agent-based simulation model. The MC-PDP [1, 3] differs from those proposed so far in the literature in a number of relevant aspects:

(i) It focuses on crime prevention rather than detention. For this reason it does not consider the emergency calls but it is based on a crime risk estimation. (The crime risk estimation can be obtained by a Predictive Policing model, as illustrated by the authors in [3].) Previous approaches [18, 23] optimize the reaction time to crime incidents, that is, crimes that have already happened.

(ii) It optimizes at the same time attributes of area, crime risk, compactness, and support. In particular, mutual support is a novel attribute that differs from backup coverage [22] in that the former regards the possibility of receiving backing in any point of the patrol sector from any other agent in the district, while the latter only concerns the overlapping areas between patrol sectors.

(iii) It considers the decision-maker's preferences in the objective function. In the formulation proposed by D'Amico et al. [18], the users can specify their preferences only by adjusting the righthand side coefficients in the constraints, while, in the models presented by Curtin et al. [21, 22], no user preference is considered.

(iv) It requires a limited amount of data to function, while all the approaches previously presented in the literature require specific information, such as the time, location, and service time of incidents and emergency calls, which might not be available. For the same reason, these methodologies do not take into consideration, and hence they cannot be extended to, all the nonviolent crimes that are not reported by emergency calls, such as pickpockets, theft of vehicles, or property damage.

In this paper we extend the first formulation of the MC-PDP [1, 3], solving the problem on a generic graph rather than on a grid. This improvement allows for the definition of patrolling configurations using census districts as the atomic unit of patrolling which results in an increased operationality of the patrolling configurations designed by the problem, a simpler access to demographic data, and favors communication with other agencies. To translate the MC-PDP into a generic graph we devised a definition of set convexity specifically tailored to the structure of the problem. In terms of methodology, we propose and compare three local search algorithms that are capable of generating good patrolling configurations in a short time. In fact, the MC-PDP has been designed to be included in a Decision Support System. Therefore, the ability of generating good patrolling configurations in a short time is of primary importance. In Section 5 we show that the solutions obtained by two of the new algorithms outperform those of the former approach. All these improvements greatly enhance the realism, applicability, and effectiveness of the MC-PDP, compared to the previous formulation and solution methodology [1, 3].

The MC-PDP is also related to another family of problems, namely, the Convex p-Partition Problem. The Convex p-Partition Problem concerns the partition of a graph into p convex subgraphs. Research on the decision form of the problem is extremely recent and the first contributions on the topic are due to Artigas et al. In two subsequent articles [24, 25] the authors prove that the problem of deciding if a graph can be partitioned into p convex subgraphs is $\mathscr{NP}$-complete in general and polynomial for cographs. In recent years, it has been shown that this problem is also polynomial for bipartite graphs for all $p \geq 2$ [26], and for planar graphs when $p = 2$ [27]. However, determining whether the problem is polynomial for planar graph and $p \geq 3$ (which is the premise of the MC-PDP) is still an open question.

So far, the research on the problem has focused primarily on its decision form (meaning establishing whether a graph can be partitioned into convex subsets). To the best of the authors' knowledge, the literature has a lack of models that tackle the problem of optimizing convex partitions, that is, partitioning a graph into convex subgraphs while optimizing an objective function or a set of criteria. In fact, optimization of convex partitions is a prerogative of the districting problem, a special case of the graph partitioning problem to which the MC-PDP belongs, and most of the research in the area focused on proposing metrics of convexity for districting problems (see, e.g., [28–30]). With respect to the Police Districting Problem, the only model that includes a measure of graph convexity in the optimization process, apart from the MC-PDP, is that proposed by D'Amico et al. [18]. However, the authors recognize that their definition of convexity "is somewhat unclear." In fact, instead of relying on the formal definition of convexity as we do for the MC-PDP, they consider a set of feasibility constraints that, according to the authors, should ensure convexity in the final solution. Nonetheless, the constraints seem to be rather arbitrary and no formal proof is given.

In the following section we formulate the MC-PDP for a generic graph and propose a methodology to deal with the problem of partitioning a generic graph into convex subsets.

3. The Multicriteria Police Districting Problem on Graph

The MC-PDP concerns the design of patrol sector configurations that are efficient and that distribute the workload homogeneously among the police officers. A solution to the MC-PDP defined on graph $G = (N, E)$ is partition P of set of nodes N. Each block $A \in P$ of the partition is a connected subset of the node set and represents a patrol sector. Therefore, from this point onward the terms "partition block," "patrol sector," and "sector" will be used interchangeably. The MC-PDP requires the partition blocks to be convex. This condition has been introduced to ensure that all the patrol sector would be intrinsically efficient; that is, the agent can move within the sector always following the shortest path. Finally, the number of subsets in the partition must be exactly p. The formal elements of the model are presented in the following.

3.1. Data and Properties. We define the MC-PDP on a generic graph $G = (N, E)$, with N being the set of nodes and E the set of edges. For each node $i \in N$ the following data is required:

(i) $a_i \in \mathbb{R}_{\geq 0}$: total length of the streets to be patrolled at node $i \in N$.

(ii) $r_i \in \mathbb{R}_{\geq 0}$: risk of crime at node $i \in N$.

Also, each edge $(i, j) \in E$ is characterized by the following:

(i) $l_{ij} \in \mathbb{R}_{\geq 0}$: length of edge $(i, j) \in E$.

Finally, $p \in \mathbb{N}_{\geq 2}$ is the number of patrolling sectors to be defined.

Additionally, on the set of nodes N and all of its subsets $N' \subseteq N$ we define the following operations:

(i) $d_{i,j}(N')$: shortest path distance between nodes $i, j \in N'$ computed using only the nodes in N'. This distance is calculated considering the length of the edges in the path.

(ii) $d^1_{i,j}(N')$: shortest edge distance between nodes $i, j \in N'$ computed using only the nodes in N'. This distance is calculated considering exclusively the number of edges in the path.

Given a node subset $N' \subseteq N$, the shortest distances between all the nodes are obtained using the Floyd-Warshall algorithm [31, 32]. The algorithm is initialized with l_{ij} for $d_{i,j}(N')$, and with the adjacency matrix for $d^1_{i,j}(N')$. Other relevant properties defined on the set of nodes N and all of its subsets $N' \subseteq N$ are as follows:

(i) $\oslash_{N'}$: diameter of subset N'. The diameter is the maximum distance between two nodes belonging to N'; that is, $\oslash_{N'} = \max_{i,j \in N'} d_{i,j}(N')$.

(ii) $c_{N'}$: center of subset N'. We define the center of a subset of nodes $N' \subseteq N$ as the node belonging to the subset that minimizes the maximum risk-weighted distance to all the other nodes in the subset. In case of ties, the node that minimizes the sum of the risk-weighted distances is chosen. In summary, $c_{N'} = \arg \operatorname{Lex} \min_{i \in N'} (\max_{j \in N'} r_j d_{i,j}(N'), \sum_{j \in N'} r_j d_{i,j}(N'))$, where Lex stands for lexicographic optimization (i.e., hierarchical optimization) of the two objectives. We consider risk-weighted distances as we assume that the agents should spend more time patrolling the nodes having greater risk.

3.2. Patrol Sector Attributes and Workload. The MC-PDP evaluates the patrol sectors defined by a partition according to four main attributes: area, isolation, demand, and diameter. All the attributes, explained in the following, are expressed as dimensionless ratios, so as to be comparable.

(i) Area, α^A. This attribute is a measure of the size of the territory that an agent should patrol. It is expressed as the ratio of the area encompassed by patrol sector A to the whole district area:

$$\alpha^A = \frac{\sum_{i \in A} a_i}{\sum_{i \in N} a_i}. \tag{1}$$

(ii) Isolation, β^A. In the MC-PDP, two patrol sectors support each other if the distance between their centers is less than or equal to a defined constant, K. The value of K can be provided by an expert. Alternatively, for the MC-PDP on graph we recommend the following:

$$K = \frac{\oslash_N}{2\sqrt{p}}; \tag{2}$$

that is, we suggest K to be set equal to the total diameter of the graph divided by twice the square root of the number of subsets to be defined. The support received by a patrol sector can be calculated by

$$b^A = \left| \left\{ B \in P \mid d_{c_A c_B}(N) \le K,\ A \ne B \right\} \right|; \tag{3}$$

that is, the support b^A is equal to the number of sectors whose centers are within a distance of K from the center of the currently considered subset. Therefore, the isolation of sector A is computed as

$$\beta^A = \frac{p - 1 - b^A}{p - 1}. \tag{4}$$

(iii) Risk, γ^A. This attribute is a measure of the total risk associated with the sector that an agent patrols. It is expressed as the ratio of the total risk of sector A to the whole district risk:

$$\gamma^A = \frac{\sum_{i \in A} r_i}{\sum_{i \in N} r_i}. \tag{5}$$

(iv) Diameter, δ^A. The diameter has been introduced in the MC-PDP as an efficiency measure. In fact, the diameter can be interpreted as the maximum distance that the agent associated with the sector would have to travel in case of an emergency call. Therefore, a small diameter results in a low response time. The diameter measure used to evaluate a patrol sector is the ratio of the subset diameter to the diameter of the graph, that is, the maximum possible diameter:

$$\delta^A = \frac{\oslash_A}{\oslash_N}. \tag{6}$$

The decision-maker can express their preference on each attribute by defining a normalized vector of weights $\mathbf{w} \in \mathbb{R}^4$. By linearly combining the attributes with preference weights $\mathbf{w}$ we can compute a measure of workload W^A of a sector A as

$$W^A = w_\alpha \cdot \alpha^A + w_\beta \cdot \beta^A + w_\gamma \cdot \gamma^A + w_\delta \cdot \delta^A. \tag{7}$$

3.3. Objective Function. The objective of the MC-PDP is to generate patrolling configurations that are efficient and, at the same time, that distribute the workload homogeneously among the patrol sectors. The objective function of the MC-PDP takes into consideration the preferences of the decision-maker for these factors by introducing coefficient $\lambda \in \mathbb{R}$, $0 \le \lambda \le 1$, that expresses the decision-maker's preference between optimization and workload balance:

$$\operatorname{obj}(P) = \lambda \cdot \max_{A \in P} \left\{ W^A \right\} + (1 - \lambda) \cdot \frac{\sum_{A \in P} W^A}{p}. \tag{8}$$

The term $\max_{A \in P}\{W^A\}$ represents the worst workload, while the term $\sum_{A \in P} W^A / p$ is the average workload. This objective function allows the decision-maker to examine the trade-off between optimization and balance by a parametric analysis. In fact, by varying λ, the model gives a range from optimization ($\lambda = 0$) to balance ($\lambda = 1$).

3.4. Problem Formulation. We can now present a mathematical formulation for the MC-PDP:

$$\begin{aligned}
\min \quad & \operatorname{obj}(P) && (9)\\
\text{s.t.} \quad & \emptyset \notin P && (10)\\
& \bigcup_{A \in P} A = N && (11)\\
& A \cap B = \emptyset \quad \forall A, B \in P \mid A \ne B && (12)\\
& |P| = p && (13)\\
& \operatorname{Conn}(A) = 1 \quad \forall A \in P && (14)\\
& \operatorname{Conv}(A) = 1 \quad \forall A \in P. && (15)
\end{aligned}$$

The model optimizes the objective function (8). Constraints (10)–(12) represent the conditions held by partition P defined on N; that is, P should not contain the empty set $\emptyset$ (10) and the partition blocks cover N (11) and are pairwise disjoint (12). Restriction (13) concerns the partition cardinality and enforces the number of partition blocks to be exactly p. Conditions (14) and (15) regard the geometry of the patrol sectors. In fact, Conn(A) is an indicator function that equals 1 when A is connected and zero otherwise, and Conv(A) is an indicator function that equals 1 when A is convex and zero otherwise. The model establishes that only connected partition blocks are feasible. This condition implies that an agent cannot be assigned to patrol sectors composed of two or more separate areas of the city. Furthermore, all the partition blocks are required to be convex. When a subset is convex,

```
procedure $LocalSearch(P_0)$
$P^* \leftarrow P_0$; {Initialize the best solution found to the initial solution.}
$t \leftarrow 0$;
while $\neg TerminationCriteria()$ do
   $P_{t+1} \leftarrow SelectNeighbor(P_t)$; {Select a neighboring solution.}
   if $P_{t+1}$ better than $P^*$ then
      $P^* \leftarrow P_{t+1}$; {Save the best solution found so far.}
   end if
   $t \leftarrow t + 1$; {Increase the iteration counter.}
end while
return $P^*$;
end procedure
```

Algorithm 1: Local search algorithm pseudocode.

it is also optimally efficient in terms of distance between the points. In fact, in a convex subset there is a minimal shortest path connecting any pair of points. Therefore, this condition allows for the generation of patrol sectors that are more efficient in terms of movement inside of the area. In the following, we illustrate more in detail the concept of graph convexity.

3.5. A Note on Graph Convexity and on Convex Graph Partitioning. Let $G = (N, E)$ be a finite simple graph. Let $A \subseteq N$, its closed interval $I[A]$ being the set of all nodes lying on shortest paths between any pair of nodes of A. The set A is convex if $I[A] = A$. In this work, the following equivalent condition is applied:

$$d^1_{i,j}(A) = d^1_{i,j}(N) \quad \forall i, j \in A \Longleftrightarrow \operatorname{Conv}(A) = 1. \tag{16}$$

Lemma 1. *Equation (16) is a proper condition for set convexity.*

Proof. Let A be a nonconvex set. It follows from the definition that $I[A] \neq A$. Let us consider nodes $i, j \in A$ and node $k \in N$ such that $k \in I[A]$ and $k \notin A$. It follows that $d^1_{i,j}(A) > d^1_{i,j}(N)$. In fact, if it were that $d^1_{i,j}(A) = d^1_{i,j}(N)$ then k would need to belong to A. Now let A be a convex set. It follows from the definition that $I[A] = A$. More specifically, all the nodes lying on the shortest path in $i, j \in A$ also belong to A. It follows that, necessarily, $d^1_{i,j}(A) = d^1_{i,j}(N)$. □

Artigas et al. [25] prove that the problem of deciding if a graph can be partitioned into $p \geq 2$ convex sets is $\mathcal{NP}$-complete. As we do not make any assumption on graph G, convexity for all the patrol sectors could not always be possible. In order to always obtain a solution, we relax constraint (15) and penalize its violation in the objective function by means of a Lagrange multiplier. The resulting program is

$$\min \quad \overline{\operatorname{obj}}(P) = \operatorname{obj}(P) + \mu \sum_{A \in P} (1 - \operatorname{Conv}(A)) \tag{17}$$

$$\text{s.t.} \quad (10), (11), (12), (13), \text{ and } (14). \tag{18}$$

Coefficient μ is the Lagrange multiplier associated with the convexity constraint (15). We suggest setting $\mu > 1$. In fact, as $\operatorname{obj}(P) \leq 1$, setting $\mu > 1$ translates into always preferring a convex graph partition over a nonconvex one, regardless of the value of $\operatorname{obj}(P)$.

4. Local Search Methods for the MC-PDP

Local search algorithms move from solution to solution in the space of candidate solutions (the search space) by applying local changes, until certain termination criteria are satisfied: for example, a solution deemed optimal is found or a time bound is elapsed. One of the main advantages of local search algorithms is that they are *anytime algorithms*, which means that they can return a valid solution even if they are interrupted at any time before they end. For this reason, they are often used to tackle hard optimization problems in a real-time environment, such as the MC-PDP. All the local search algorithms proposed make use of the same solution structure:

$$P = \{d_1, d_2, \ldots, d_{|N|}\}, \tag{19}$$

where $d_i \in [1, p]$, $\forall i = 1, \ldots, N$. In summary, a solution is a vector of $|N|$ elements, one for each node in the graph, that can take any value from one to p, that is, the number of patrolling sectors. A generic pseudocode for a local search algorithm is presented in Algorithm 1.

The procedure starts the search from a given initial solution P_0 and it iteratively moves to a solution belonging to the neighborhood of the incumbent one, until certain termination criteria are met. The neighborhood of a solution is the set of solutions that can be obtained from the current one by changing it slightly. In this research, we consider all the solutions that can be obtained by removing a node from a patrol sector and assigning it to another one, without violating constraints (10)–(14). Different implementations of *TerminationCriteria*() and *SelectNeighbor*() result in different local search algorithms. The characteristics of the algorithms developed in this research are presented in the following.

Simple Hill Climbing. At each iteration, the Simple Hill Climbing (SHC) algorithm [33] explores the neighborhood of the incumbent solution to find a better one. In our SHC, *SelectNeighbor*(P_t) procedure explores the neighborhood

```
procedure MultiStart()
while ¬TerminationCriteria() do
    $P \leftarrow$ InitialSolution(); {Generate an initial solution.}
    $P' \leftarrow$ LocalSearch($P$); {Improve the current solution.}
    if $P'$ better than $P^*$ then
        $P^* \leftarrow P'$; {Save the best solution found so far.}
    end if
end while
return $P^*$;
end procedure
```

Algorithm 2: Multistart pseudocode.

of partition P_t in a random fashion and returns the first improving solution found. The algorithm terminates when no improving solution is found or the time limit is exceeded.

Steepest Descent Hill Climbing. The Steepest Descent Hill Climbing (SDHC) algorithm [33] is a variant of the SHC that explores the whole neighborhood of the incumbent solution and chooses the best solution belonging to it. This is the same algorithm originally proposed for the solution of the MC-PDP [1].

Tabu Search. Similarly to the SDHC, the Tabu Search (TS) algorithm [34, 35] explores the whole neighborhood of the incumbent solution. However, the TS chooses for the next iteration the best solution found that is not *tabu*. Also, the TS does not terminate if an improving solution is not found. This allows the algorithm to escape local optima. The criterion that is used to declare a certain point of the neighborhood as tabu is based on a short-term memory. At each iteration, the TS presented in this paper stores the current solution in the short-term memory with an associated expiration counter initially set to T. During the exploration of a neighborhood all the solutions found that are already included in the short-term memory are marked as tabu and their expiration counter is reset to T. Finally, at the end of the iteration, all the expiration counters are decreased by one and the solutions whose counters have reached zero are removed from the short-term memory.

The algorithm terminates when the time limit is exceeded, when no nontabu solution is found in the current neighborhood, or after a fixed number I of nonimproving iterations. We suggest setting parameters T and I to the cardinality of the node set; that is, $T = I = |N|$.

4.1. Multistart Local Search Algorithms. Local search methods are very good at exploring certain zones of the solution space but they generally end up in local optima. Multistart is a very simple and general diversification method. In order to better explore distant portions of the solution space the search is started more than one time from different points. The pseudocode of a multistart procedure is illustrated in Algorithm 2.

The procedure alternates a solution generation procedure with a local search step, until the time limit is exceeded.

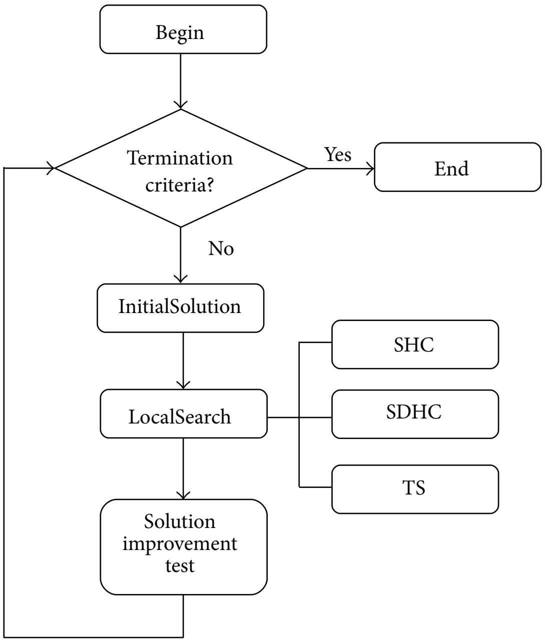


Figure 1: Algorithm flow-chart.

Generating an Initial Solution. To generate an initial solution at each iteration of the multistart algorithm, we use the random greedy algorithm proposed in Camacho-Collados et al. [1], adapted to work on a generic graph. In summary, the algorithm generates a solution by randomly choosing the first node of each sector and then expanding the sectors in a greedy fashion while preserving their connectivity. Initially, the partition blocks are empty. In the first phase of the algorithm, each block is initialized with a randomly chosen node. Subsequently, at each iteration of the second phase, the algorithm extends the initial solution by assigning a node to a single sector. The algorithm chooses the combination of node and sector that results in the best feasible solution. The algorithm ends when all the points have been assigned to subsets. It is important to notice that, in the current version of the algorithm, the solutions are evaluated by using the relaxed objective function (17).

Complete Algorithm Structure. Figure 1 illustrates the flowchart of the complete multistart algorithm. First, an initial solution is generated using the aforementioned random greedy algorithm. Second, the solution is improved by means of local search. The local search algorithm could be either SHC, SDHC, or TS. Finally, the incumbent solution is compared with the best found that is updated if required. This sequence of steps is repeated until the termination criteria are met, the algorithm terminates, and the best solution found is returned.

5. Results and Discussion

5.1. Dataset. We test our algorithms on the Central District of Madrid dataset presented in Camacho-Collados et al. [1]. However, in this research the data is aggregated with respect

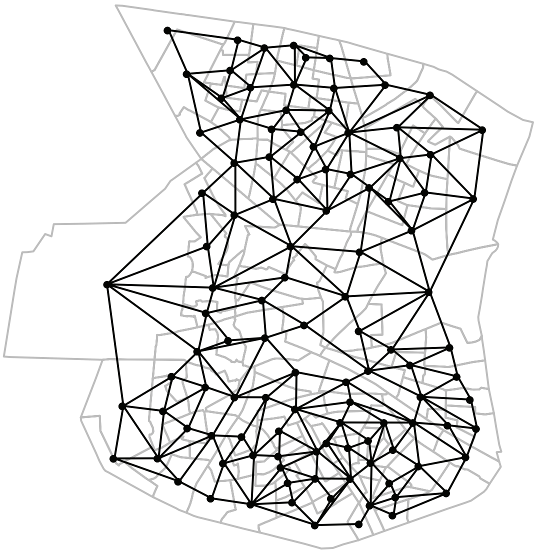


Figure 2: Census districts in the Central District of Madrid (in gray) and the corresponding graph (in black).

to the census district rather than a grid. As reported by Sarac et al. [4] the use of a structure based on census districts is preferable as it allows easy access to demographic data and is suitable for use by other agencies. Figure 2 shows the subdivision of the territory and the associated graph. The borders of the census districts are plotted in gray. The nodes of the graph, identified by black bullets, correspond to the centroids of the census districts. Finally, black lines represent the edges of the graph that connect neighboring census districts. Overall, the graph is comprised of 111 nodes and 277 edges. The total length of the streets at each node, a_i, is obtained by summing the length of the parts of street contained within the borders of each census district. The length of each edge, l_{ij}, is computed as the great-circle distance between the nodes. In terms of the risk of crime at each node, r_i, we consider the thefts that occurred during the following shifts:

(i) SATT3: Saturday, 10/13/2012, night shift (10 PM–8 AM).

(ii) SUNT1: Sunday, 10/14/2012, morning shift (8 AM–3 PM).

(iii) MONT2: Monday, 10/15/2012, afternoon shift (3 PM–10 PM).

These shifts have been identified by a service coordinator in charge of the patrolling operations of the Central District of Madrid as typical scenarios representing different crime activity patterns, as illustrated in Figure 3. In the SATT3 shift the district is characterized by a high level of nightlife; therefore thefts are committed in almost all the territory, with the highest levels distributed around popular meeting places in the center and in the northeast of the district. SUNT1 has a low level of criminality, mostly concentrated in the south of the district where a popular flea market is held every Sunday morning. Finally, MONT2 presents the characteristics of a normal business day, with criminal activity spread in the central area of the territory, which is where the commercial activities are located.

5.2. Computational Experiments. We now present and analyze the solution values obtained by the heuristic algorithms presented. It is important to notice that all the algorithms (i.e., SHC, SDHC, and TS) implement the multistart method; that is, the local search algorithms are repeatedly run starting from a different randomly generated solution, until the global termination criteria are met. For the sake of consistency, the experiments have been run using the same parameters adopted in previous researches on the subject [1, 3]:

(i) Decision-maker preference weights and balance coefficient: $(w_\alpha, w_\beta, w_\gamma, w_\delta) = (0.45, 0.05, 0.45, 0.05)$ and $\lambda = 0.1$. These values have been provided by a service coordinator in charge of the patrolling operations of the Central District of Madrid as their preference.

(ii) Number of patrol sectors: $p = \{2, 6\}$. On an "average day," the Central District of Madrid is either split into two big sectors or partitioned according to its six neighborhoods.

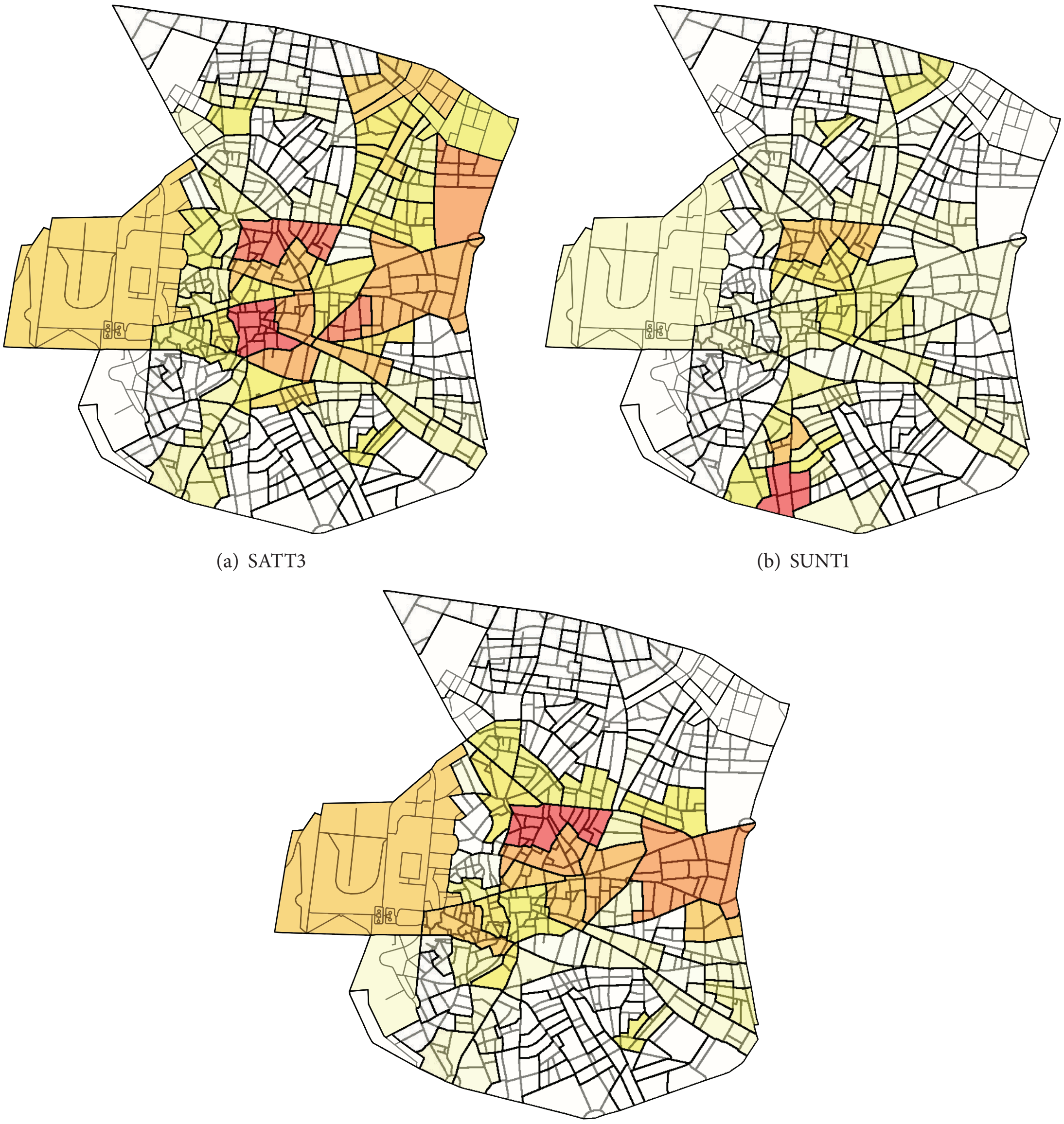
(iii) The parameters of the TS algorithm have been set as $T = I = 111$, as the graph is comprised of 111 nodes.

Given the random nature of the algorithms proposed, we ran each combination of algorithm, shift, and number of patrol sectors 50 times. Each run had a time limit of 60 seconds, to simulate the real-time environment of DSS. The experiments were run on a computer with an Intel Core i5-2500K CPU having four cores at 3.30 GHz and 4 GB RAM memory and the algorithms were programmed in C++.

Tables 1(a)–1(f) show the average relaxed objective function value, $\overline{\text{obj}}(P)$, and the corresponding standard deviation for each group. In the tables, the rows correspond to the algorithm and the best average solution value is highlighted in bold. Please note that a solution value that is less than one indicates that the solution is feasible with respect to the convexity constraints (15). From the tables we can observe that on average the TS algorithm finds the best solution in four out of six groups and the SDHC in the remaining two groups.

For the sake of reproducibility and comparison with future contributions to the subject, the average number of multistart iterations (The number of solutions explored in a full local search run depends exclusively on the initial solution and, in the case of the SHC, on the neighborhood exploration order which is random. Therefore, the only metric that fully depends on the capabilities of the machine is the number of multistart iterations. Reporting its value allows for comparisons between runs on different machines.) and the corresponding standard deviations are provided in Tables 2(a)–2(f).

5.3. Statistical Analysis. As a first step, we verify that the solution values are normally distributed by using the Shapiro-Wilk test. The results are shown in Table 3. As all the p values are greater than the chosen alpha level, 0.05, the null



(a) SATT3

(b) SUNT1

(c) MONT2

Figure 3: Maps of the number of thefts reported in the Central District of Madrid. The red shade represents a high crime level while the white shade represents no criminal activity.

hypothesis that the solution values are normally distributed cannot be rejected. To understand if the differences in the means are statistically significant we can now run one-way ANOVA tests. The results are illustrated in Table 4.

We highlighted in bold the rows of the groups where a significant difference was detected. We can immediately see that there is no significant difference in the groups where the SDHC algorithm was the best. We run post hoc Tukey's tests to understand more in detail which algorithm performs better for the solution of the MC-PDP. Tukey's test is a single-step multiple comparison procedure used to find means that are significantly different from each other and that is more suitable for multiple comparisons than doing a number of t-tests would be. The results are illustrated in Tables 5(a)–5(f). In the tables, the rows are associated with the pairs of algorithms being tested. We highlighted in bold the rows showing a significant difference. From the results of the statistical tests we can draw the following conclusions:

(i) The performances of the SHC and the SDHC in terms of solutions' objective function values are always identical, except for shift SATT3 with six patrol sectors, where the SDHC produces solutions that are significantly better than those of the SHC.

(ii) The TS produces on average better solutions in four out of six groups, and its performances are not worse than those of the other two algorithms in the remaining two groups. Therefore, we can claim that it is preferable to use the TS over the SHC and the SDHC.

Table 1: Average relaxed objective function value, $\overline{\text{obj}}(P)$, and standard deviation for each group.

(a) Shift SATT3, $p = 2$

Algorithm	Avg.	St. Dev.
SHC	0.50109	0.00435
SDHC	**0.49997**	**0.00413**
TS	0.53567	0.19831

(b) Shift SATT3, $p = 6$

Algorithm	Avg.	St. Dev.
SHC	0.20720	0.00342
SDHC	0.20498	0.00313
TS	**0.20146**	**0.00513**

(c) Shift SUNT1, $p = 2$

Algorithm	Avg.	St. Dev.
SHC	0.50456	0.01000
SDHC	0.50651	0.01277
TS	**0.49101**	**0.00473**

(d) Shift SUNT1, $p = 6$

Algorithm	Avg.	St. Dev.
SHC	0.20619	0.00315
SDHC	0.20594	0.00328
TS	**0.20161**	**0.00384**

(e) Shift MONT2, $p = 2$

Algorithm	Avg.	St. Dev.
SHC	0.50381	0.00608
SDHC	**0.50067**	**0.00656**
TS	0.51948	0.14180

(f) Shift MONT2, $p = 6$

Algorithm	Avg.	St. Dev.
SHC	0.20350	0.00469
SDHC	0.20336	0.00498
TS	**0.19729**	**0.00620**

Table 2: Average number of multistart iterations and corresponding standard deviation for each group.

(a) Shift SATT3, $p = 2$

Algorithm	Avg.	St. Dev.
SHC	65.88	8.73800
SDHC	70.04	8.90531
TS	5.15094	0.36142

(b) Shift SATT3, $p = 6$

Algorithm	Avg.	St. Dev.
SHC	64.54	10.44951
SDHC	55.7	10.12826
TS	6.58	0.73095

(c) Shift SUNT1, $p = 2$

Algorithm	Avg.	St. Dev.
SHC	57.74	7.01546
SDHC	53.54	7.25121
TS	5.68	0.68333

(d) Shift SUNT1, $p = 6$

Algorithm	Avg.	St. Dev.
SHC	92.54	15.46188
SDHC	54.66	7.85223
TS	5.38	0.49031

(e) Shift MONT2, $p = 2$

Algorithm	Avg.	St. Dev.
SHC	62.44	8.68769
SDHC	59	7.65522
TS	5.3	0.46291

(f) Shift MONT2, $p = 6$

Algorithm	Avg.	St. Dev.
SHC	80.02	11.29654
SDHC	63.26	11.74875
TS	11.02	0.99980

5.4. Solution Analysis. Figures 4(a)–4(f) illustrate the best solutions found for each shift and number of patrol sectors in terms of relaxed objective function value. All the solutions have been identified by the TS. In the figures, the borders of the census districts have been plotted in black, the streets have been plotted in gray, and each patrol sector is represented by a different color. By observing the patrolling configurations some insights can be drawn:

(i) SATT3: police activity is focused mostly on the center, as well as on the northeast part of the district, where most of the crimes are committed. The reason for that is that those areas are very busy nightlife meeting places.

(ii) SUNT1: the patrolling configurations concentrate on the southern part of the territory, where most of the thefts happen on Sunday morning because of the popular flea market. In the six patrol sectors' configuration we can see that one sector is dedicated exclusively to the area with the highest concentration of crimes that corresponds exactly to the location of the flea market.

(iii) MONT2: the district is uniformly partitioned between northeast and southwest. The configuration with six patrol sectors assigns higher importance to the central-western part of the district, corresponding to the commercial area.

6. Conclusions

In this paper we extended the MC-PDP to generate efficient convex partitions on generic graphs, which increases the practical usefulness and applicability of the model. Also, we propose and compare three local search algorithms and test

Table 3: Values of Shapiro-Wilk test statistics and corresponding p value for each group.

(a) Shift SATT3, $p = 2$

Algorithm	Statistic W	p value
SHC	0.97584	0.06277
SDHC	0.98543	0.3412
TS	0.98595	0.3707

(b) Shift SATT3, $p = 6$

Algorithm	Statistic W	p value
SHC	0.96566	0.1534
SDHC	0.97822	0.4797
TS	0.95943	0.08413

(c) Shift SUNT1, $p = 2$

Algorithm	Statistic W	p value
SHC	0.98348	0.2456
SDHC	0.97758	0.08592
TS	0.98024	0.1387

(d) Shift SUNT1, $p = 6$

Algorithm	Statistic W	p value
SHC	0.96181	0.1059
SDHC	0.97153	0.2669
TS	0.9772	0.08032

(e) Shift MONT2, $p = 2$

Algorithm	Statistic W	p value
SHC	0.97593	0.06385
SDHC	0.97819	0.09604
TS	0.97526	0.05652

(f) Shift MONT2, $p = 6$

Algorithm	Statistic W	p value
SHC	0.97063	0.2454
SDHC	0.98254	0.663
TS	0.98201	0.19

Table 4: Results of the one-way ANOVA tests on the solution values.

Shift	p	$F(2, 147)$	Pr(>F)
SATT3	2	1.57	0.212
SATT3	**6**	**26.2**	**1.85e − 10**
SUNT1	**2**	**37.44**	**7.19e − 14**
SUNT1	**6**	**28.16**	**4.43e − 11**
MONT2	2	0.754	0.472
MONT2	**6**	**22.12**	**4.01e − 09**

them on real crime data from the Central District of Madrid. The results of the computational experiments show that the TS presented in this paper produces solutions that are on average better than those identified by the SDHC algorithm proposed in a previous research [1].

Table 5: Results of Tukey's test for each group.

(a) Shift SATT3, $p = 2$

Pair	p value
SHC-SDHC	0.99867
TS-SDHC	0.26697
TS-SHC	0.28945

(b) Shift SATT3, $p = 6$

Pair	p value
SHC-SDHC	**0.01698**
TS-SDHC	**6.09e − 5**
TS-SHC	**<1e − 7**

(c) Shift SUNT1, $p = 2$

Pair	p value
SHC-SDHC	0.57937
TS-SDHC	**<1e − 7**
TS-SHC	**<1e − 7**

(d) Shift SUNT1, $p = 6$

Pair	p value
SHC-SDHC	0.93070
TS-SDHC	**<1e − 7**
TS-SHC	**<1e − 7**

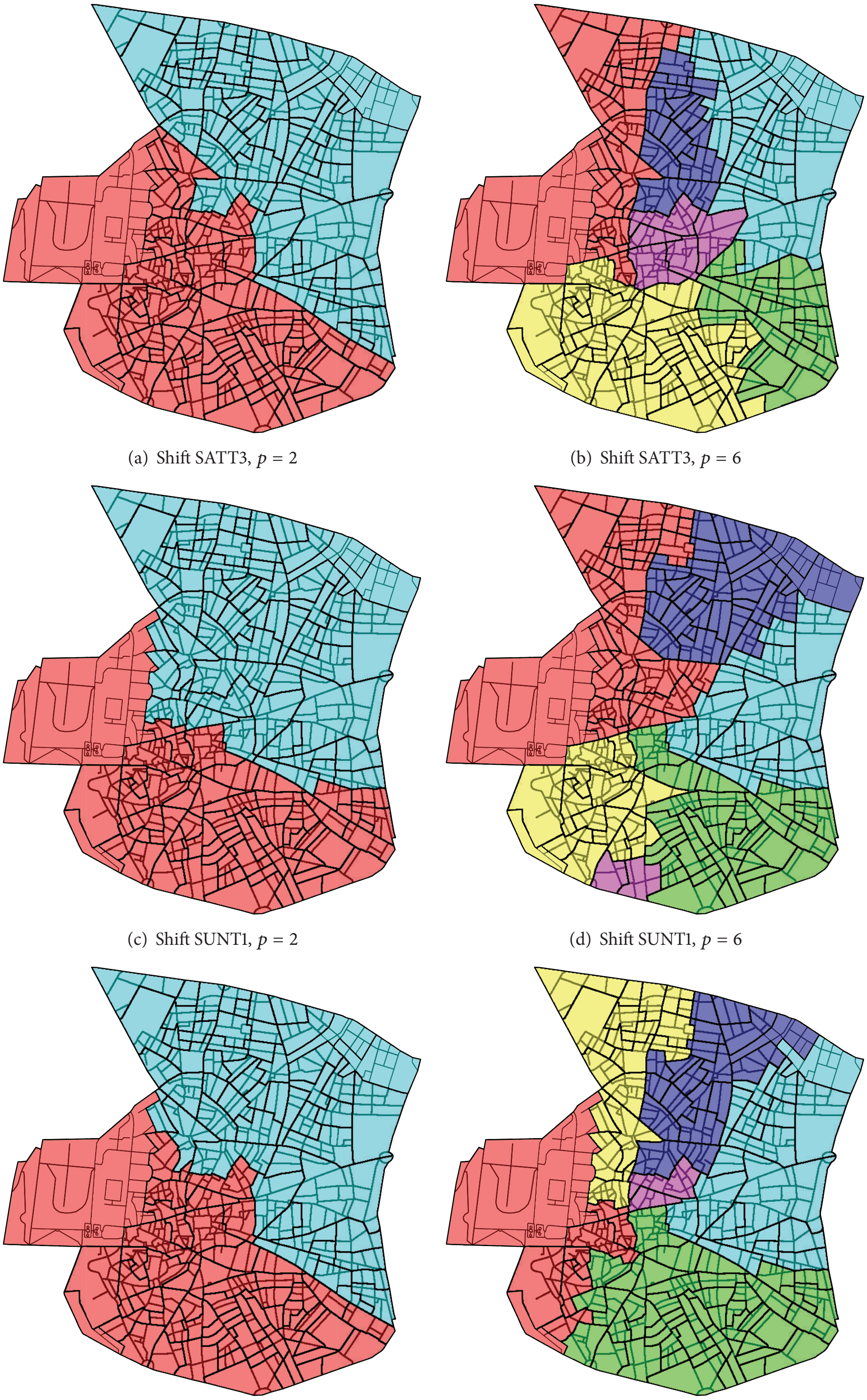
(e) Shift MONT2, $p = 2$

Pair	p value
SHC-SDHC	0.98003
TS-SDHC	0.48723
TS-SHC	0.60639

(f) Shift MONT2, $p = 6$

Pair	p value
SHC-SDHC	0.99018
TS-SDHC	**2e − 7**
TS-SHC	**1e − 7**

This research offers new interesting lines to be pursued. In terms of modeling, solving the MC-PDP on a graph simplifies the inclusion of demographic data in the model, such as the racial composition of a census district. A future work might explore the impact of the minimization of racial profiling on the performance of the resulting patrolling configurations. Furthermore, it would be interesting to research optimal solution algorithms for the MC-PDP. Given its intrinsically nonlinear structure and the interdependence between the patrol sectors to compute the isolation ratio, we believe that a simultaneous column-and-row generation algorithm [36] should be used. Although this methodology is rather time consuming, it could still generate good heuristic solutions within the allowed time limit. The quality of these solutions could then be compared with that of the TS proposed in this paper. As explained in Introduction, the MC-PDP is a variant of the graph partitioning problems. It could be interesting



(a) Shift SATT3, $p = 2$

(b) Shift SATT3, $p = 6$

(c) Shift SUNT1, $p = 2$

(d) Shift SUNT1, $p = 6$

(e) Shift MONT2, $p = 2$

(f) Shift MONT2, $p = 6$

Figure 4: Best solutions found.

to compare our TS with adaptations of solution techniques proposed in the literature for this family of problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the Spanish National Police Corps personnel of the SEYCO (Statistics and Operative Control Section), the CPD (Data Processing Center), the Sala 091 of Madrid (Police Emergency Call Center), the Central District of Madrid, and the Central District of Granada for their help and collaboration. The investigation of Camacho-Collados was partially financed by the Spanish Police Foundation research grant for civil servants of the Spanish National Police Corps and by the Fulbright Program. The research of Liberatore was supported by the Spanish Government, Grant TIN2012-32482. All support is gratefully acknowledged.

References

[1] M. Camacho-Collados, F. Liberatore, and J. M. Angulo, "A multi-criteria police districting problem for the efficient and effective design of patrol sector," *European Journal of Operational Research*, vol. 246, no. 2, pp. 674–684, 2015.

[2] W. L. Perry, B. McInnis, C. C. Price, S. C. Smith, and J. S. Hollywood, *Predictive Policing. The Role of Crime Forecasting in Law Enforcement Operations*, RAND Corporation, Santa Monica, Calif, USA, 2013.

[3] M. Camacho-Collados and F. Liberatore, "A decision support system for predictive police patrolling," *Decision Support Systems*, vol. 75, pp. 25–37, 2015.

[4] A. Sarac, R. Batta, J. Bhadury, and C. Rump, "Reconfiguring police reporting districts in the city of Buffalo," *OR Insight*, vol. 12, no. 3, pp. 16–24, 1999.

[5] D. He and Y. L. Hong, "An improved tabu search algorithm based on grid search used in the antenna parameters optimization," *Mathematical Problems in Engineering*, vol. 2015, Article ID 947021, 8 pages, 2015.

[6] X. H. Zhang, S. Q. Zhong, Y. L. Liu, and X. L. Wang, "A framing link based tabu search algorithm for large-scale multidepot vehicle routing problems," *Mathematical Problems in Engineering*, vol. 2014, Article ID 152494, 13 pages, 2014.

[7] G. Lin, W. Zhu, and M. M. Ali, "A tabu search-based memetic algorithm for hardware/software partitioning," *Mathematical Problems in Engineering*, vol. 2014, Article ID 103059, 15 pages, 2014.

[8] Y. Z. Yang and X. S. Gu, "Cultural-based genetic tabu algorithm for multiobjective job shop scheduling," *Mathematical Problems in Engineering*, vol. 2014, Article ID 230719, 14 pages, 2014.

[9] E. Rolland, H. Pirkul, and F. Glover, "Tabu search for graph partitioning," *Annals of Operations Research*, vol. 63, pp. 209–232, 1996.

[10] B. Ucar, C. Aykanat, K. Kaya, and M. Ikinci, "Task assignment in heterogeneous computing systems," *Journal of Parallel and Distributed Computing*, vol. 66, no. 1, pp. 32–46, 2006.

[11] U. Benlic and J.-K. Hao, "An effective multilevel memetic algorithm for balanced graph partitioning," in *Proceedings of the 22nd International Conference on Tools with Artificial Intelligence (ICTAI '10)*, pp. 121–128, Arras, France, October 2010.

[12] U. Benlic and J.-K. Hao, "A multilevel memetic approach for improving graph k-partitions," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 624–642, 2011.

[13] U. Benlic and J.-K. Hao, "An effective multilevel tabu search approach for balanced graph partitioning," *Computers and Operations Research*, vol. 38, no. 7, pp. 1066–1075, 2011.

[14] P. Galinier, Z. Boujbel, and M. Coutinho Fernandes, "An efficient memetic algorithm for the graph partitioning problem," *Annals of Operations Research*, vol. 191, no. 1, pp. 1–22, 2011.

[15] P. S. Mitchell, "Optimal selection of police patrol beats," *The Journal of Criminal Law, Criminology, and Police Science*, vol. 63, no. 4, article 577, 1972.

[16] S. E. Bodily, "Police sector design incorporating preferences of interest groups for equality and efficiency," *Management Science*, vol. 24, no. 12, pp. 1301–1313, 1978.

[17] R. Benveniste, "Solving the combined zoning and location problem for several emergency units," *Journal of the Operational Research Society*, vol. 36, no. 5, pp. 433–450, 1985.

[18] S. J. D'Amico, S.-J. Wang, R. Batta, and C. M. Rump, "A simulated annealing approach to police district design," *Computers & Operations Research*, vol. 29, no. 6, pp. 667–684, 2002.

[19] J. M. Chaiken and P. Dormont, "A patrol car allocation model: background," *Management Science*, vol. 24, no. 12, pp. 1280–1290, 1978.

[20] J. M. Chaiken and P. Dormont, "A patrol car allocation model: capabilities and algorithms," *Management Science*, vol. 24, no. 12, pp. 1291–1300, 1978.

[21] K. Curtin, F. Qui, K. Hayslett-McCall, and T. Bray, "Geographic information systems and crime analysis," in *Integrating GIS and Maximal Coverage Models to Determine Optimal Police Patrol Areas*, pp. 214–235, Idea Group, 2005.

[22] K. M. Curtin, K. Hayslett-McCall, and F. Qiu, "Determining optimal police patrol areas with maximal covering and backup covering location models," *Networks and Spatial Economics*, vol. 10, no. 1, pp. 125–145, 2010.

[23] Y. Zhang and D. E. Brown, "Police patrol districting method and simulation evaluation using agent-based model & GIS," *Security Informatics*, vol. 2, article 7, 2013.

[24] D. Artigas, M. C. Dourado, and J. L. Szwarcfiter, "Convex partitions of graphs," *Electronic Notes in Discrete Mathematics*, vol. 29, pp. 147–151, 2007.

[25] D. Artigas, S. Dantas, M. C. Dourado, and J. L. Szwarcfiter, "Partitioning a graph into convex sets," *Discrete Mathematics*, vol. 311, no. 17, pp. 1968–1977, 2011.

[26] L. N. Grippo, M. Matamala, M. D. Safe, and M. J. Stein, "Convex *p*-partitions of bipartite graphs," *Theoretical Computer Science*, vol. 609, part 2, pp. 511–514, 2016.

[27] R. Glantz and H. Meyerhenke, "Finding all convex cuts of a plane graph in cubic time," in *Algorithms and Complexity*, vol. 7878 of *Lecture Notes in Computer Science*, pp. 246–263, Springer, 2013.

[28] J. Bozeman, L. Pyrik, and J. Theoret, "Nearly convex sets and the shape of legislative districts," *International Journal of Pure and Applied Mathematics*, vol. 49, no. 3, pp. 341–347, 2008.

[29] C. P. Chambers and A. D. Miller, "A measure of bizarreness," *Quarterly Journal of Political Science*, vol. 5, no. 1, pp. 27–44, 2010.

[30] J. K. Hodge, E. Marshall, and G. Patterson, "Gerrymandering and convexity," *The College Mathematics Journal*, vol. 41, no. 4, pp. 312–324, 2010.

[31] R. W. Floyd, "Algorithm 97: shortest path," *Communications of the ACM*, vol. 5, no. 6, artilce 345, 1962.

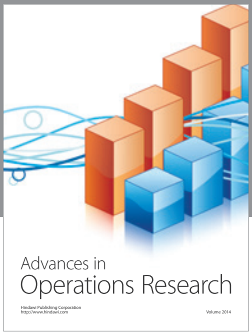
[32] S. Warshall, "A theorem on boolean matrices," *Journal of the Association for Computing Machinery*, vol. 9, pp. 11–12, 1962.

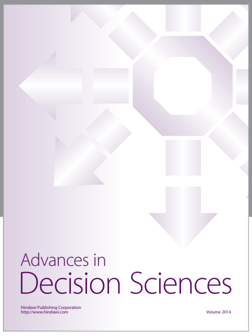
[33] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Pearson, 3rd edition, 2009.

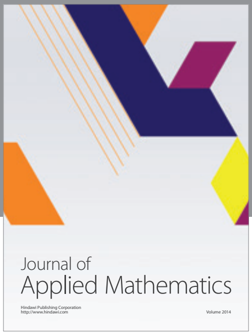
[34] F. Glover, "Tabu search—part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

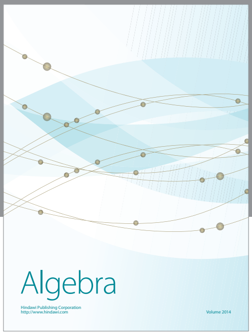
[35] F. Glover, "Tabu search—part II," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4–32, 1990.

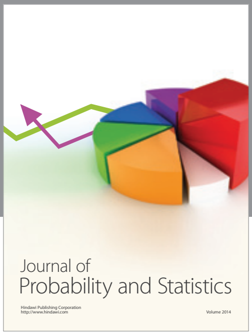
[36] İ. Muter, Ş. İlker Birbil, and K. Bülbül, "Simultaneous column-and-row generation for large-scale linear programs with column-dependent-rows," *Mathematical Programming*, vol. 142, no. 1-2, pp. 47–82, 2013.


Advances in
Operations Research

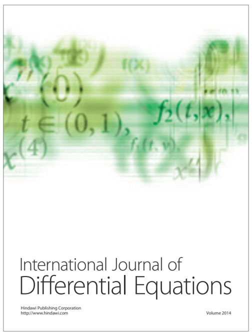

Advances in
Decision Sciences


Journal of
Applied Mathematics

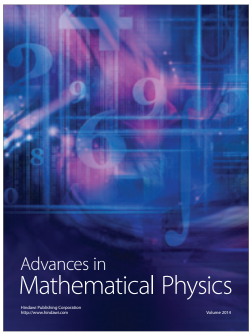

Algebra

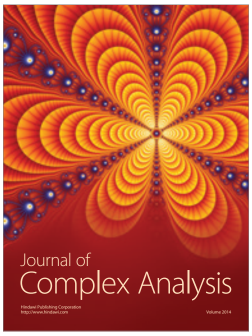

Journal of
Probability and Statistics

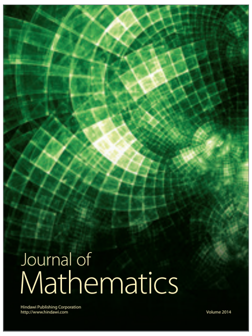
The Scientific
World Journal

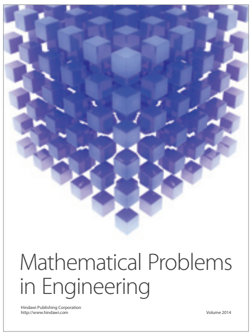

International Journal of
Differential Equations

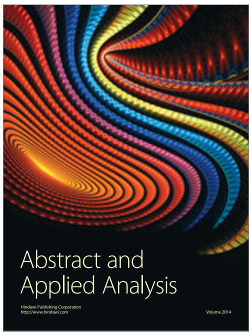
International Journal of
Combinatorics

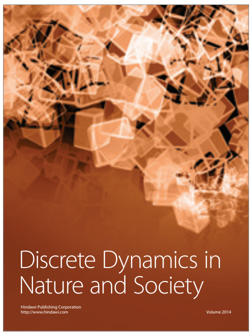

Advances in
Mathematical Physics

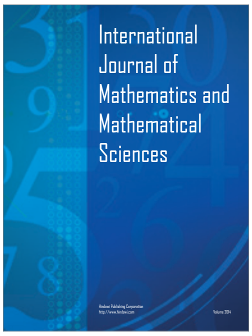

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

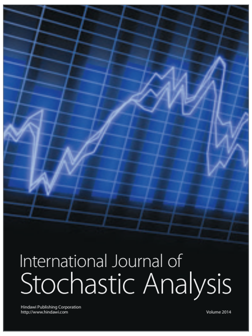

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization