A Comparison of Local Search Methods for the Multi-Criteria Police Districting Problem on Graph

F. Liberatore∗, M. Camacho-Collados†

May 28, 2015

Abstract

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 Multi-Criteria Police Districting Problem (MC-PDP) concerns the definition of sound patrolling sectors in a police district. The model was originally formulated in collaboration with the Spanish National Police Corps and was solved by means of a Steepest Descent Hill Climbing algorithm. One of the major limitations of the MC-PDP is that it requires the territory to be organized as a grid. In this work we formulate the MC-PDP for a generic graph, which results in a more applicable and usable model. Also, we propose for its solution three local search algorithms, including a Tabu Search. 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 those of the previously used algorithm. 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. [3].

∗ Corresponding author.
AIDA, Computer Science Department, Autonomous University of Madrid, Madrid, Spain.
federico.liberatore@urjc.es

† Deputy Directorate for Operations, Spanish National Police Corps, Madrid, Spain.
Statistics and Operations Research Department, University of Granada, Granada, Spain.
mcamacho0007@policia.es
The newest member of this family is the Multi-Criteria Police Districting Problem (MC-PDP) [3].

The novelty of this model stands in that it evaluates the workload associated to 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 [10], the MC-PDP allows to design 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 [2] that presented a Decision Support System (DSS) for the implementation of a paradigm of Predictive Patrolling in the SNPC.

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, as well as propose and compare new heuristic algorithms. More specifically, the original MC-PDP was formulated to partition a grid. In this article 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. [12] 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. 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 [7], vehicle routing [15], hardware/software partitioning [8] and job shop scheduling [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. In Section 2 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 4 we explain the dataset and the computational experiments ran to test the algorithms. Also, we analyze the results and provide insights on the solutions obtained. Finally, we conclude the article with some guidelines for future research.

2 The Multi-Criteria 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 a graph \( G = (N, E) \) is a partition \( P \) of the 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, i.e., 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.

2.1 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:

- \( a_i \in \mathbb{R}_{\geq 0} \): Total length of the streets to be patrolled at node \( i \in N \).
- \( r_i \in \mathbb{R}_{\geq 0} \): Risk of crime at node \( i \in N \).

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

- \( l_{ij} \in \mathbb{R}_{\geq 0} \): 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:
• \(d_{ij}(N')\): Shortest path distance between nodes \(i, j \in N'\) computed using only the nodes in \(N'\).

• \(d_{1ij}(N')\): Shortest edge distance between nodes \(i, j \in N'\) computed using only the nodes in \(N'\).

Given a node subset \(N' \subseteq N\), the shortest distances between all the nodes are obtained using the Floyd-Warshall algorithm \([4, 13]\). The algorithm is initialized with \(l_{ij}\) for \(d_{ij}(N')\), and with the adjacency matrix for \(d_{1ij}(N')\). Other relevant properties defined on the set of nodes \(N\) and all of its subsets \(N' \subseteq N\) are:

• \(\varnothing_{N'}\): Diameter of subset \(N'\). The diameter is the maximum distance between two nodes belonging to \(N'\), i.e., \(\varnothing_{N'} = \max_{i,j \in N'} d_{ij}(N')\).

• \(c_{N'}\): 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, \(c_{N'} = \arg \text{Lex} \min_{i \in N'} \left( \max_{j \in N'} r_j d_{ij}(N') , \sum_{j \in N'} r_j d_{ij}(N') \right)\), where \(\text{Lex}\) stands for Lexicographic 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.

2.2 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.

• **Area**, \(\alpha^A\): 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.

\[
\alpha^A = \frac{\sum_{i \in A} a_i}{\sum_{i \in N} a_i}
\]  

(1)

• **Isolation**, \(\beta^A\): 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:

\[ K = \frac{\mathcal{G}_N}{2\sqrt{p}} \]  

(2)

The support received by a patrol sector can be calculated by:

\[ b^A = |\{ B \in P | d_{c,Acb} (N) \leq K, \ A \neq B \}| \]  

(3)

Therefore, the isolation of sector $A$ is computed as:

\[ \beta^A = \frac{p - 1 - b^A}{p - 1} \]  

(4)

- **Risk**, $\gamma^A$: This attribute is a measure of the total risk associated to the sector that an agent patrols. It is expressed as the ratio of the total risk of sector $A$ to the whole district risk.

\[ \gamma^A = \frac{\sum_{i \in A} r_i}{\sum_{i \in N} r_i} \]  

(5)

- **Diameter**, $\delta^A$: 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 to 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.

\[ \delta^A = \frac{\mathcal{G}_A}{\mathcal{G}_N} \]  

(6)

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

\[ W^A = w_\alpha \cdot \alpha^A + w_\beta \cdot \beta^A + w_\gamma \cdot \gamma^A + w_\delta \cdot \delta^A \]  

(7)
2.3 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 the coefficient $\lambda \in \mathbb{R}$, $0 \leq \lambda \leq 1$ that expresses the decision maker’s preference between optimization and workload balance.

$$\min \ obj(P) = \lambda \cdot \max_{A \in P} \{W^A\} + (1 - \lambda) \cdot \frac{\sum_{A \in P} W^A}{p} \quad (8)$$

The term $\max_{A \in P} \{W^A\}$ represents the worst workload, while the term $\frac{\sum_{A \in P} W^A}{p}$ 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 $\lambda$, the model gives a range from optimization ($\lambda = 0$) to balance ($\lambda = 1$).

2.4 Problem Formulation

We can now present a mathematical formulation for the MC-PDP:

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

The model optimizes the objective function (8). The constraints (10)-(12) represent the conditions held by a partition $P$ defined on $N$: $P$ should not contain the empty set $\emptyset$ (10), the partition blocks cover $N$ (11) and are pairwise disjoint (12). The restriction (13) concerns the partition cardin-
nality 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 a 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, 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.

### 2.5 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 $I[A]$ is 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.

$$d_{i,j}^1(A) = d_{i,j}^1(N) \forall i, j \in A \iff Conv(A) = 1$$

(16)

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

*Proof.* Let $A$ be a non convex set. It follows from the definition that $I[A] \neq A$. Let us consider nodes $i, j \in A$ and a node $k \in N$ such that $k \in I[A]$ and $k \notin A$. It follows that $d_{i,j}^1(A) > d_{i,j}^1(N)$. In fact, if it were that $d_{i,j}^1(A) = d_{i,j}^1(N)$ 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 between between $i, j \in A$ also belong to $A$. It follows that, necessarily, $d_{i,j}^1(A) = d_{i,j}^1(N)$. □

Artigas et al. [1] prove that the problem of deciding if a graph can be partitioned into $p \geq 2$ convex sets is $NP$-complete. As we do not make any assumption on the 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:
\[
\min \quad \overline{\text{obj}}(P) = \text{obj}(P) + \mu \sum_{A \in P} (1 - \text{Conv}(A)) \\
\text{s.t.} \quad (10) - (14)
\]

The coefficient \( \mu \) is the Lagrange multiplier associated to the convexity constraint (15). We suggest setting \( \mu > 1 \). In fact, as \( \text{obj}(P) \leq 1 \), setting \( \mu > 1 \) translates into always preferring a convex graph partition over a non-convex one, regardless of the value of \( \text{obj}(P) \).

### 3 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, e.g., 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. A generic pseudocode for a local search algorithm is presented in Algorithm 1.

#### Algorithm 1 Local Search algorithm pseudocode.

```plaintext
procedure LocalSearch(P₀)
    P* ← P₀; \{Initialize the best solution found to the initial solution.\}
    t ← 0;
    while ¬TerminationCriteria() do
        P_{t+1} ← SelectNeighbor(P_t); \{Select a neighboring solution.\}
        if P_{t+1} better than P* then
            P* ← P_{t+1}; \{Save the best solution found so far.\}
        end if
        t ← t + 1; \{Increase the iteration counter.\}
    end while
    return P*;
end procedure
```

The procedure starts the search from a given initial solution \( P₀ \) and it iteratively moves to a solution belonging to the neighborhood of the incumbent one, until a certain termination criteria is met. Different implementations of \( \text{TerminationCriteria()} \) and \( \text{SelectNeighbor()} \) result in dif-
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 [11] explores the neighborhood of the incumbent solution to find a better one. 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). In our SHC, the $\text{SelectNeighbor}(P_t)$ procedure explores the neighborhood of the 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 [11] 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 [3].

**Tabu Search**

Similarly to the SDHC, the Tabu Search (TS) algorithm [5, 6] explores the whole neighborhood of the incumbent solution. However, the TS chooses for the next iteration the best solution found that is not *tabu* and it 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 that are already included in the short term memory are marked as tabu and their expiration counter are 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 non-tabu solution is found in the current neighborhood, or after a fixed number $I$ of non-improving iterations. We suggest setting the parameters $T$ and $I$ to the cardinality of the node set, i.e., $T = I = |N|$.

### 3.1 Multi-Start 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. Multi-start 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 multi-start procedure is illustrated in Algorithm 2.

**Algorithm 2** Multi-start pseudocode.

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

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

**Generating an Initial Solution**

To generate an initial solution at each iteration of the multi-start algorithm, we use the random greedy algorithm proposed in Camacho-Collados et al. [3], 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).

4 Results and Discussion

4.1 Dataset

We test our algorithm on the Central District of Madrid dataset presented in Camacho-Collados et al. [3]. However, in this research the data is aggregated with respect to the census district rather than a grid. As reported by Sarac et al. [12] 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 1 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 occurred during the following shifts:

- SATT3: Saturday, 10/13/2012, night shift (10 PM–8 AM).
- SUNT1: Sunday, 10/14/2012, morning shift (8 AM–3 PM).
- 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 2. 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 north-east 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
Figure 1: Census districts in the Central District of Madrid (in gray) and the corresponding graph (in black).
Figure 2: 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.

(a) SATT3  
(b) SUNT1  
(c) MONT2
criminal activity spread in the central area of the territory, that is where the commercial activities are located.

4.2 Computational Experiments

The experiments have been run using the same parameters adopted in previous researches on the subject [3, 2].

- 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 her preference.

- 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.

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 a DSS. The experiments were run on a computer with an Intel Core i5-2500K CPU having four cores at 3.30GHz and 4GB RAM memory and the algorithm were programmed in C++.

Tables 1a-1f show the average relaxed objective function value, \( \overline{obj}(P) \), 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.

4.3 Statistical Analysis

To understand if the differences in the means are statistically significant we run one-way ANOVA tests. The results are illustrated in Table 2. 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
Table 1: Average relaxed objective function value, $\overline{\text{obj}}(P)$, and standard deviation for each group.

(a) Shift SATT3, $p = 2$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg.</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC</td>
<td>0.50109</td>
<td>0.00435</td>
</tr>
<tr>
<td>SDHC</td>
<td>0.49997</td>
<td>0.00413</td>
</tr>
<tr>
<td>TS</td>
<td>0.53567</td>
<td>0.19831</td>
</tr>
</tbody>
</table>

(b) Shift SATT3, $p = 6$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg.</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC</td>
<td>0.20720</td>
<td>0.00342</td>
</tr>
<tr>
<td>SDHC</td>
<td>0.20498</td>
<td>0.00313</td>
</tr>
<tr>
<td>TS</td>
<td>0.20146</td>
<td>0.00513</td>
</tr>
</tbody>
</table>

(c) Shift SUNT1, $p = 2$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg.</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC</td>
<td>0.50456</td>
<td>0.01000</td>
</tr>
<tr>
<td>SDHC</td>
<td>0.50651</td>
<td>0.01277</td>
</tr>
<tr>
<td>TS</td>
<td>0.49101</td>
<td>0.00473</td>
</tr>
</tbody>
</table>

(d) Shift SUNT1, $p = 6$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg.</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC</td>
<td>0.20619</td>
<td>0.00315</td>
</tr>
<tr>
<td>SDHC</td>
<td>0.20594</td>
<td>0.00328</td>
</tr>
<tr>
<td>TS</td>
<td>0.20161</td>
<td>0.00384</td>
</tr>
</tbody>
</table>

(e) Shift MONT2, $p = 2$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg.</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC</td>
<td>0.50381</td>
<td>0.00608</td>
</tr>
<tr>
<td>SDHC</td>
<td>0.50067</td>
<td>0.00656</td>
</tr>
<tr>
<td>TS</td>
<td>0.51948</td>
<td>0.14180</td>
</tr>
</tbody>
</table>

(f) Shift MONT2, $p = 6$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg.</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC</td>
<td>0.20350</td>
<td>0.00469</td>
</tr>
<tr>
<td>SDHC</td>
<td>0.20336</td>
<td>0.00498</td>
</tr>
<tr>
<td>TS</td>
<td>0.19729</td>
<td>0.00620</td>
</tr>
</tbody>
</table>

Table 2: Results of the one-way ANOVA tests on the solution values.

<table>
<thead>
<tr>
<th>Shift</th>
<th>$p$</th>
<th>$F(2, 147)$</th>
<th>$\Pr(&gt; F)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SATT3</td>
<td>2</td>
<td>1.57</td>
<td>0.212</td>
</tr>
<tr>
<td>SATT3</td>
<td>6</td>
<td>26.2</td>
<td>1.85e-10</td>
</tr>
<tr>
<td>SUNT1</td>
<td>2</td>
<td>37.44</td>
<td>7.19e-14</td>
</tr>
<tr>
<td>SUNT1</td>
<td>6</td>
<td>28.16</td>
<td>4.43e-11</td>
</tr>
<tr>
<td>MONT2</td>
<td>2</td>
<td>0.754</td>
<td>0.472</td>
</tr>
<tr>
<td>MONT2</td>
<td>6</td>
<td>22.12</td>
<td>4.01e-09</td>
</tr>
</tbody>
</table>
Table 3: Results of Tukey’s test for each group.

(a) Shift SATT3, $p = 2$.

<table>
<thead>
<tr>
<th>Pair</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC-SDHC</td>
<td>0.99867</td>
</tr>
<tr>
<td>TS-SDHC</td>
<td>0.26697</td>
</tr>
<tr>
<td>TS-SHC</td>
<td>0.28945</td>
</tr>
</tbody>
</table>

(b) Shift SATT3, $p = 6$.

<table>
<thead>
<tr>
<th>Pair</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC-SDHC</td>
<td>0.01698</td>
</tr>
<tr>
<td>TS-SDHC</td>
<td>6.09e-5</td>
</tr>
<tr>
<td>TS-SHC</td>
<td>&lt;1e-7</td>
</tr>
</tbody>
</table>

(c) Shift SUNT1, $p = 2$.

<table>
<thead>
<tr>
<th>Pair</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC-SDHC</td>
<td>0.57937</td>
</tr>
<tr>
<td>TS-SDHC</td>
<td>&lt;1e-7</td>
</tr>
<tr>
<td>TS-SHC</td>
<td>&lt;1e-7</td>
</tr>
</tbody>
</table>

(d) Shift SUNT1, $p = 6$.

<table>
<thead>
<tr>
<th>Pair</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC-SDHC</td>
<td>0.93070</td>
</tr>
<tr>
<td>TS-SDHC</td>
<td>&lt;1e-7</td>
</tr>
<tr>
<td>TS-SHC</td>
<td>&lt;1e-7</td>
</tr>
</tbody>
</table>

(e) Shift MONT2, $p = 2$.

<table>
<thead>
<tr>
<th>Pair</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC-SDHC</td>
<td>0.98003</td>
</tr>
<tr>
<td>TS-SDHC</td>
<td>0.48723</td>
</tr>
<tr>
<td>TS-SHC</td>
<td>0.60639</td>
</tr>
</tbody>
</table>

(f) Shift MONT2, $p = 6$.

<table>
<thead>
<tr>
<th>Pair</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHC-SDHC</td>
<td>0.99018</td>
</tr>
<tr>
<td>TS-SDHC</td>
<td>2e-7</td>
</tr>
<tr>
<td>TS-SHC</td>
<td>1e-7</td>
</tr>
</tbody>
</table>

A single-step multiple comparison procedure was 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 3a-3f. 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:

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

- The TS produces on average better solutions in four out of six groups, and its performances are not worst 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.

4.4 Solution Analysis

Figures 3a-3f 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 in gray and each patrol
Figure 3: Best solutions found.

(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, $p = 6$. 
sector is represented by a different color. By observing the patrolling configurations some insights can be drawn:

- **SATT3**: Police activity is focused mostly on the center, as well as on the north-east 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.

- **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.

- **MONT2**: The district is uniformly partitioned between north-east and south-west. The configuration with six patrol sectors assigns higher importance to the central-western part of the district, corresponding to the commercial area.

5 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 algorithm and test them on real crime data from the Central District of Madrid. The results of the computational experiments show that the TS presented in this article produces solutions that are on average better than those identified by the SDHC algorithm proposed in a previous research [3].

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 non-linear structure and the interdependence between the patrol sectors to compute the isolation ratio, we believe that a simultaneous column-and-row generation algorithm [9] 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.

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 Liberatorre was supported by the Spanish Government, grant TIN2012-32482. All support is gratefully acknowledged.

References


