*Journal of Internet Services and Applications, 2026, 17:1,* doi: 10.5753/jisa.2026.6805


# Optimizing Preventive Police Patrol Routes in Brazilian Cities Using a GRASP-Based Routing Heuristic

**Isabel Rosseti** [ **Universidade Federal Fluminense** | ***rosseti@ic.uff.br*** ]
**Eduardo Canellas** [ **Universidade Federal Fluminense** | ***eduardocanellas@id.uff.br*** ]
**Wagner Santos** [ **Polícia Militar do Estado do Rio de Janeiro** | ***wagnergs@id.uff.br*** ]
**Marcos Lage** [ **Universidade Federal Fluminense** | ***mlage@ic.uff.br*** ]
**Yuri Frota** [ **Universidade Federal Fluminense** | ***yuri@ic.uff.br*** ]
**Daniel de Oliveira** [ **Universidade Federal Fluminense** | ***danielcmo@ic.uff.br*** ]

*Institute of Computing, Universidade Federal Fluminense, Av. Gal. Milton Tavares de Souza, s/n, São Domingos, Niterói, RJ, 24210-590, Brazil.*

**Received:** 03 November 2025 • **Accepted:** 11 May 2026 • **Published:** 19 August 2026

**Abstract** Criminal activity is a major challenge for governments in large urban centers worldwide. Within Smart City initiatives, ensuring urban safety is a key priority. However, developing and applying data-driven approaches to reduce urban crime remains an open, yet important, issue. This paper addresses the Routing of Police Vehicles in Large Urban Centers (`RVP-Urb`) problem, which aims to maximize crime coverage while considering resource constraints. The proposed solution, called `PROTECTOR`, is a heuristic based on the Greedy Randomized Adaptive Search Procedure (GRASP), designed to generate preventive patrol routes that enhance police visibility in crime hotspots. Unlike existing methods, `PROTECTOR` mirrors the structure of Brazilian police departments by dividing urban areas into smaller zones and generating both intra- and inter-zone routes. The heuristic uses real-world crime hotspot data over defined time intervals to create adaptive patrol plans that can be updated as hotspot patterns change. Experimental results using the `RolRoute-DS` dataset demonstrate that `PROTECTOR` performs effectively, producing near-optimal routes comparable to those generated by exact mathematical models while maintaining computational efficiency.

**Keywords:** Preventive Policing, Preventive Police Patrol Routes, GRASP, Metaheuristic



## 1 Introduction

Criminality is a multifaceted and complex problem that affects many large nations worldwide [García-Ponce and Laterzo, 2023; Borrion *et al.*, 2019; Eysenck and Gudjonsson, 2013]. According to the statistics reported by Numbeo[1], several countries in 2025 continued to show high crime rates, often influenced by a combination of social and economic factors. Thus, several initiatives within the context of Smart Cities [Coelho *et al.*, 2019; Bilal *et al.*, 2020] have focused on mitigating criminal activity, as the concept of a smart city inherently encompasses a safe and secure urban environment [Chen *et al.*, 2017; Lourenço *et al.*, 2018; Ristvej *et al.*, 2020; Dewinter *et al.*, 2020; Samanta *et al.*, 2022]. Nevertheless, determining how to explore existing technologies to reduce urban criminality effectively remains an open issue.

When analyzing Numbeo's Crime Index (computed by dividing the number of *officially* reported crimes by a city's total population and multiplying the result by 100,000) at the city level, it becomes clear that several metropolitan areas, including national capitals, continue to face challenges related to crime. Despite the existence of several smart city initiatives specifically designed to reduce criminal activity in cities such as Rio de Janeiro and Johannesburg [Silva *et al.*, 2017; Lourenço *et al.*, 2018; Obagbuwa and Abidoye, 2021], these urban areas still present comparatively higher crime indices than other cities impacted by wars, *e.g.*, Baghdad.

To illustrate the multifaceted challenges of implementing smart city strategies to mitigate crime activity, let us examine the cases of Rio de Janeiro and São Paulo, two Brazilian cities. These two urban centers will serve as examples throughout the paper. Rio de Janeiro is the second most populous city in Brazil and ranks sixth in the Americas. São Paulo is not only the most populous city in Brazil but also the largest in South America and the Southern Hemisphere. Economically, São Paulo plays an essential role in the national context, with its Gross Domestic Product (GDP) accounting for approximately 10.7% of Brazil's total GDP[2].

Given their socioeconomic importance, both Rio de Janeiro and São Paulo have experienced sustained inflows of migrants, primarily driven by individuals seeking employment and better living conditions. This movement has been further intensified by the prolonged economic crisis that has affected Brazil over the last decade. Many of these migrants, however, tend to settle in socially and spatially segregated

[1] `https://www.numbeo.com/common/about.jsp`

[2] `https://www.ibge.gov.br/`

urban peripheries, *i.e.*, regions characterized by precarious infrastructure, limited public services, and pre-existing high levels of violence [Dos Anjos Júnior *et al.*, 2018]. Consequently, the expansion of these vulnerable areas increases the population exposed to criminal activities and urban insecurity [Hjalmarsson *et al.*, 2024]. According to data from the Public Security Institute of the State of Rio de Janeiro[3], the number of street robberies has followed an irregular yet upward trajectory since 2017. A similar trend can be observed in homicide rates for the same period[4], revealing a persistent pattern of urban violence despite ongoing governmental and community-level interventions.

Due to increased criminal activity in these cities, public security authorities at various levels of government have intensified efforts to develop tools using existing urban data to support the formulation of effective public policies to reduce violence. These initiatives often involve measures to prevent criminal events before they occur. This proactive approach to law enforcement is commonly referred to as *Preventive Policing* [Alikhademi *et al.*, 2022]. The central idea behind preventive policing is to make police officers' presence and actions more visible and perceptible to the general public, either by deploying officers at strategic locations throughout the city or by conducting regular patrols. Within the context of preventive policing, one of the most complex operational tasks to be addressed is the *Routing of Police Vehicles in Large Urban Centers* (`RVP-Urb`), as discussed by Saint-Guillain *et al.* [2021].

The inherent complexity of the `RVP-Urb` problem lies in the fact that available resources are typically limited, *i.e.*, the number of police vehicles and officers is often considerably smaller than the number required to ensure adequate coverage of the entire city. Consequently, police officers, who are generally responsible for defining patrol routes, must determine which specific routes each vehicle should follow. In practice, these routing decisions tend to prioritize the so-called Hotspots [Reis *et al.*, 2006], *i.e.*, urban areas with significantly higher crime rates than other parts of the city.

To address the `RVP-Urb` problem, it is necessary to maximize the coverage of areas with the highest crime rates within a given region through the generation of optimized patrol routes. The generation of such routes must take into account several operational constraints, as defined by the police authorities: (i) the number of available police units (*e.g.*, patrol vehicles, officers, police stations, *etc.*); (ii) the specific regions of the city to be considered (*i.e.*, urban zones); (iii) the maximum route distance allowed for each type of police unit; and (iv) the city's topology, which can be represented as a graph composed by several street segments.

Thus, a patrol route, in the context of `RVP-Urb`, is always modeled as a cyclic path along the city's street graph and can be classified into two categories: (i) *Type 1*: routes that may traverse more than one zone but are subject to a maximum distance constraint (*i.e.*, inter-zone routes). Each inter-zone route typically includes a set of points of interest (*e.g.*, schools, banks) that must be mandatorily covered; and (ii) *Type 2*: routes that must remain entirely within a single zone and are also bounded by a maximum allowable distance (*i.e.*, intra-zone routes). It is important to note that this zoning definition and route classification are operational strategies commonly adopted by the Military Police across several Brazilian states. In addition to these mobile patrol routes, fixed units can also be strategically positioned throughout the city (*e.g.*, officers at permanent posts or police bases).

The `RVP-Urb` problem is NP-complete [Lenstra and Rinnooy Kan, 1981], implying that exact mathematical formulations quickly become computationally intractable for medium- or large-scale urban scenarios. As a result, the adoption of heuristic or metaheuristic solution approaches becomes not only advantageous but necessary to obtain solutions that are both practically feasible and close to optimal. In this context, this manuscript introduces `PROTECTOR`, a heuristic based on the Greedy Randomized Adaptive Search Procedure (GRASP) [Resende and Ribeiro, 2016], specifically designed to generate preventive police patrol routes in Brazilian urban environments and to effectively tackle the `RVP-Urb` problem. The performance and applicability of the proposed heuristic are evaluated using real-world data from the city of São Paulo. The experimental results demonstrate the effectiveness of automated, heuristic-driven solutions for the `RVP-Urb` problem, illustrating their potential to substantially improve police operational planning, resource allocation, and coverage in complex and populated cities.

The remainder of this manuscript is structured as follows. Section 2 provides background and discusses key concepts in police patrol. Section 3 presents a discussion of related work. Section 4 formally defines the `RVP-Urb` problem and presents its mathematical formulation, establishing the foundation for the proposed solution. Section 5 describes the `PROTECTOR` heuristic in detail, outlining its design principles and algorithmic components. Section 6 presents and analyzes the experimental evaluation, highlighting the performance and effectiveness of the proposed approach on real-world data. Finally, Section 7 concludes the paper and outlines directions for future research.

# 2 Background Knowledge

This section presents the basic concepts of preventive police patrolling that underpin this paper. The definitions and operational principles discussed herein are grounded in the official frameworks adopted by the Police Departments of Rio de Janeiro and São Paulo. In both metropolitan regions, the urban territory is divided into smaller areas, a practice designed to facilitate more efficient management, coordination, and resource allocation.

## 2.1 A Brief Tour Through Criminal Analysis

Police departments typically engage in a wide range of activities based on methodologies and concepts from multiple fields of knowledge, commonly referred to as *Criminal Analysis* [Boba, 2005]. This multidisciplinary framework encompasses three dimensions: (i) Strategic, (ii) Tactical, and (iii) Administrative. *Administrative Criminal Analysis* focuses on the application of statistics to analyze the behavior and

[3] `http://www.isp.rj.gov.br`
[4] `https://www.ispgeo.rj.gov.br`

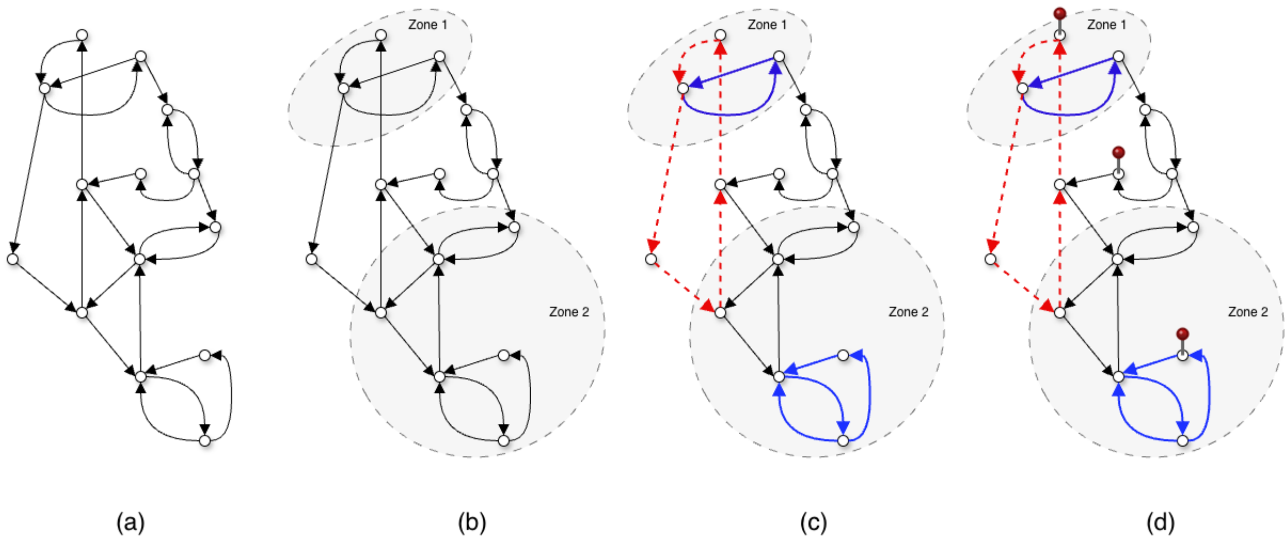


**Figure 1.** (a) Graph representation of the city's topology based on street segments. (b) Division of the urban area into zones for planning preventive police patrols. (c) Definition of three patrol routes, two intra-zone and one inter-zone. (d) Allocation of fixed police officers across selected locations.

evolution of crime-related indicators over a defined period (*i.e.*, time series analysis). By comparing data across different time windows, this approach aims to identify patterns in these indicators and produce insights. The knowledge derived from such studies informs a wide range of stakeholders, *e.g.*, citizens, public managers, governmental agencies, international organizations, and non-governmental entities.

*Tactical Criminal Analysis* is concerned with generating knowledge that supports both preventive and investigative policing activities. Within the scope of preventive policing, analytical reports are developed to identify geographic areas with elevated crime rates, determine criminals' *modus operandi*, and plan preventive patrols. Finally, *Strategic Criminal Analysis* is dedicated to producing long-term insights to understand criminal phenomena.

In any form of criminal analysis, the investigation begins with the extraction and organization of data from various sources. These sources may include relational databases, spreadsheets, textual reports, maps, and other structured or unstructured files. Once collected, the data are subjected to a range of analytical procedures, *e.g.*, statistical modeling, probabilistic reasoning, and geoprocessing techniques, with the main goal of identifying criminal patterns that may inform policing strategies.

According to Boba [2005], criminal activities can be categorized into several patterns according to the spatial and temporal dynamics of crime: (i) *Series* – a category encompassing multiple similar crimes against one or more victims or targets over a specific period; (ii) *Spree* – a succession of crimes occurring with such frequency that they appear to constitute a continuous sequence, generally committed by the same criminal or group; (iii) *Hot Spot* – a geographic area that does not present persistent criminal activity but, due to specific factors, presents an anomalous and temporary rise in crime rates; (iv) *Hot Dot* – an individual, whether a criminal, a victim, or a witness, who is directly or repeatedly associated with multiple criminal events; (v) *Hot Product* – goods or commodities that become appealing to criminals due to their resale value; and (vi) *Hot Target* – locations that are frequently chosen for crimes owing to their accessibility, visibility, or economic characteristics.

Once these patterns have been identified, law-enforcement officers can develop strategic preventive patrol routes, analytical reports, and interactive dashboards to support and enhance decision-making. These spatial and temporal crime patterns are typically identified for specific geographic regions and time intervals, as criminal activity evolves across both space and time. The following subsection discusses how police organize the territory, drawing on examples from Rio de Janeiro and São Paulo, and how preventive patrol can be defined within these territories.

## 2.2 Territorial Organization and Preventive Police Patrol

The organization of police territory varies by city and its administrative framework. In this subsection, we discuss the territorial organization models adopted by the cities of Rio de Janeiro and São Paulo. The city of Rio de Janeiro adheres to the territorial framework established by the State Secretariat for Public Security, which defines the spatial hierarchy for crime monitoring and operational coordination[5]. According to this structure, the State of Rio de Janeiro is subdivided into 137 Integrated Public Security Circumscriptions (Circunscrições Integradas de Segurança Pública - CISP, in Portuguese). These circumscriptions are grouped into 39 Integrated Areas of Public Security (Áreas Integradas de Segurança Pública - AISP, in Portuguese), which, in turn, are organized into seven Integrated Regions of Public Security (Regiões Integradas de Segurança Pública - RISP, in Portuguese). In the case of the Military Police of the State of Rio de Janeiro (Polícia Militar do Estado do Rio de Janeiro - PMERJ, in Portuguese), the following hierarchical configuration was implemented: (i) Seven Area Policing Commands (Comando de Policiamento de Área - CPA, in Portuguese) were created, each corresponding to a single RISP; (ii) Thirty-

[5] www.ispdados.rj.gov.br/divisaoTerritorial.html

nine Military Police Battalions were established and grouped under these CPAs, with each battalion corresponding to one AISP; and (iii) Within each battalion's jurisdiction, there are several police stations, each corresponding to a CISP.

The city of São Paulo follows the territorial framework established by the State Secretariat for Public Security. The Military Police of the State of São Paulo (Polícia Militar do Estado de São Paulo - PMESP, in Portuguese) operates under a structured, hierarchical system comprising 23 policing commands, nine of which are located in São Paulo. Within this structure, the Capital Policing Command (Comando de Policiamento da Capital - CPC, in Portuguese) is subdivided into eight CPAs, each responsible for coordinating law enforcement and public safety activities across São Paulo's eleven administrative regions. These divisions enable the PMESP to ensure comprehensive territorial coverage, operational efficiency, and responsiveness to the diverse security challenges characteristic of a large metropolitan area. In addition to these territorial units, the PMESP also maintains a range of specialized commands, each designed to address distinct operational domains and emergency contexts. These include the Shock Policing Command (Comando de Policiamento de Choque - CPChq, in Portuguese), which is responsible for riot control and tactical operations; the Highway Policing Command (Comando de Policiamento Rodoviário - CPRv, in Portuguese), which oversees law enforcement across the state's extensive highway network; the Environmental Policing Command (Comando de Polícia Ambiental - CPAmb, in Portuguese), dedicated to the protection of natural resources and the enforcement of environmental laws; the Fire Department Command (Comando do Corpo de Bombeiros - CCB, in Portuguese), tasked with firefighting, rescue, and disaster response; the Traffic Policing Command (Comando de Policiamento de Trânsito - CPTran, in Portuguese), focused on urban traffic regulation and accident prevention; and the Airborne Radio Patrol Group Command (Grupamento de Radiopatrulha Aérea - GRPAe, in Portuguese), which provides aerial support for ground operations and rapid emergency response.

Despite differences in territorial hierarchy, it is worth noting that both cities, as well as most cities in Brazil, rely heavily on actions implemented at a finer geographic scale, particularly within the jurisdictions of local battalions, to prevent and control criminal activity. One of the most effective strategies discussed in this context is *preventive police patrolling*. Although this concept resembles policing approaches developed in the 1980s [Hale and Chapman, 1981], a period during which many of today's technologies were not yet available, recent studies continue to demonstrate its effectiveness [Turchan and Braga, 2024; Tregle *et al.*, 2025], where they show that preventive patrolling remains a valuable crime-control strategy, particularly when police resources are strategically concentrated in a small number of geographically defined areas with high levels of violence.

This approach involves police officers, on foot or in vehicles, patrolling specific areas of the city to avoid criminal activity before it occurs. A wide range of crimes can often be prevented simply by the visible presence of law enforcement, including vehicle theft, mobile phone theft, and street robbery. The main challenge in this task is determining how to allocate police officers and which specific routes each patrol vehicle should follow. This task is complex because the number of available resources, *i.e.*, officers, vehicles, and operational units, is significantly smaller than the number required to ensure complete coverage of the urban area.

Let us illustrate the problem using the scenario depicted in Figure 1. Figure 1(a) shows the graph representation of a city's street network. In this graph, each edge corresponds to either an entire street or a street segment, along with its respective traffic direction. We use segments rather than entire streets because some streets are excessively long; segmenting them facilitates the generation and optimization of patrol routes. For each street segment, a crime index is computed by counting the number of criminal incidents occurring within a predefined temporal window centered on that segment. The police officer can configure the specific time interval for this computation based on operational needs. For instance, the index may be calculated considering crime records from the last ten years, the previous month, or even the last week. In practice, shorter time intervals are preferred, as they more accurately capture the dynamics of emerging crime hotspots, which tend to fluctuate over time.

Once the crime index is calculated for all segments, the police officer defines zones of interest, *i.e.*, the areas that will serve as the primary focus for patrol routing, as the two zones illustrated in Figure 1(b). At this stage, the officer must also determine the number and type of available patrol units, distinguishing between non-drivable units (*e.g.*, officers stationed at fixed posts) and drivable units (*e.g.*, patrol vehicles). The number of patrol routes to be planned depends directly on the number of drivable units. These routes may be intra-zone, confined within a single zone and represented by blue arrows in Figure 1(c), or inter-zone, traversing multiple zones and shown as red dashed arrows. It is important to emphasize that each patrol route is modeled as a cycle within the graph, *i.e.*, the route begins and ends at the same vertex, ensuring that patrol vehicles return to their point of origin. Each patrol route may also include mandatory segments, as specified by the police officer, corresponding to locations requiring mandatory patrol coverage, *e.g.*, schools, banks, or other facilities. Finally, in Figure 1(d), fixed units are strategically positioned at selected points in the graph to provide coverage for areas with the highest crime indices, ensuring that the city's most vulnerable locations receive surveillance. Section 4 formalizes the Routing of Police Vehicles in Large Urban Centers (`RVP-Urb`) problem as a mixed integer programming problem. The following section discusses related work.

# 3 Related Work

Several studies found in the literature have proposed approaches for the problem of preventive police route generation, as stated by Samanta *et al.* [2022]; Dewinter *et al.* [2020]. However, the operational assumptions and objectives of these approaches differ substantially from those considered in the `RVP-Urb` problem addressed in this manuscript. In particular, the formulation proposed here reflects specific operational rules used in Brazilian police patrol planning, including dividing the patrol area into zones, distinguishing between interand intra-zone routes, and requiring coverage of predefined priority edges. Because of these structural differences, existing approaches cannot be applied directly to the RVP-Urb problem or fairly compared in a computational setting. Instead, they provide conceptual background and inspiration for developing new methods tailored to this specific operational context.

The study conducted by Melo *et al*. [2005] uses the concept of societal agents to simulate the operational environment of police activities and to generate patrol routes. Unlike the other approaches discussed in this section, their model defines the environment as an ecosystem composed of autonomous agents representing various entities, *e.g.*, criminals, police officers, commercial establishments, and other facilities. Within this framework, patrol routes are generated as ordered sequences of locations that each police officer must visit within a predefined time interval. Although the concept is interesting, accurately modeling complex human behavior, particularly that of criminals, remains challenging.

Reis *et al*. [2006] propose an approach named GAPatrol, which employs a multi-agent simulation framework to assist police officers in generating patrol routes. Similar to PROTECTOR, the GAPatrol system relies on identifying crime hotspots. However, unlike our proposed approach, GAPatrol primarily focuses on discovering these hotspots rather than using predefined ones, and it does not consider different types of zones within the patrol area or the assignment of police officers to fixed positions.

Chawathe [2007] model the street network through a graph-based representation, in which vertices represent intersections and edges correspond to individual street segments as in PROTECTOR. In the model by Chawathe [2007], each edge is assigned a weight reflecting the benefit of patrolling the corresponding segment. The heuristic developed by the authors explicitly incorporates these benefit values into the route generation process. While the idea of quantifying the patrol benefit of specific areas is interesting, it introduces practical challenges. In real-world scenarios, this type of metric is subjective, as it depends on complex, context-specific factors.

Chen *et al*. [2015] proposed an approach for generating patrol routes based on the ant colony optimization technique. As in several previous studies, the authors represent the city's road network as a graph. However, their analysis is restricted to a limited subset of crime categories, *i.e.*, mobile phone theft and car theft, as the main focus of their work lies on foot patrol operations rather than vehicular patrols. Moreover, their approach does not support partitioning the city into distinct zones, which limits its flexibility in large or heterogeneous urban areas. Afterward, the authors extended their approach [Chen *et al*., 2017] to introduce a Bayesian-inspired heuristic for the cooperative routing of police patrols.

Oliveira *et al*. [2015] propose an approach to generate patrol routes that maximize both police visibility and the perceived safety of citizens at key locations such as hospitals and schools. The proposed method aims to generate a set of balanced routes that cover these strategic points. To achieve this, the authors based their formulation on the multi-vehicle covering tour problem, in which one subset of locations must be visited. On the other hand, another subset must be only near the generated routes. Similar to PROTECTOR, the authors' approach is formulated as an integer programming problem. However, unlike PROTECTOR, their model does not support defining zones within the patrol area, nor does it account for different patrol route types, thereby limiting its adaptability to more complex or heterogeneous urban security contexts.

Kim *et al*. [2023] proposed a model designed to generate routes that enable rapid and efficient responses to emergency calls. Similar to PROTECTOR, their approach uses data characterizing crime hotspots to produce optimized routes. However, unlike PROTECTOR, which is oriented toward preventive policing and proactive patrol planning, their model focuses primarily on reactive response, *i.e.*, generating routes that minimize the travel time to reach the location of a crime event.

Other studies have focused on the dynamic vehicle routing problem, incorporating environmental variability to enable real-time route adjustments. For instance, Bertsimas *et al*. [1991] introduced an approach that adapts the Dynamic Traveling Salesman Problem (DTSP) to police vehicle routing, thereby enabling patrols to be reallocated dynamically as new information becomes available. In a similar vein, Gendreau *et al*. [2006] examined the maximum expected coverage relocation problem for emergency services, including police units, while Shen *et al*. [2018] explored optimization strategies for emergency response vehicles, particularly police cars and ambulances. Also, Oliveira *et al*. [2023] propose a method based on integer linear programming to estimate the shortest path of emergency vehicles to respond to climate events. The latter study emphasizes the efficient deployment of limited resources under dynamic and uncertain operational conditions, highlighting the importance of real-time adaptability in emergency management.

# 4 RVP-Urb Problem Definition

This section presents a formal definition of the Routing of Police Vehicles in Large Urban Centers (RVP-Urb) problem. It begins by describing the basic data structure that underpins the proposed approach, *i.e.*, the crime graph, which serves as the foundation for the PROTECTOR heuristic. Subsequently, the section provides a mathematical formulation of the RVP-Urb problem as a mixed-integer programming model that captures the key decision variables, operational constraints, and optimization objectives associated with police vehicle routing in large urban environments.

## 4.1 The Crime Graph

Let $G^c = (V, E, Q)$ be a graph, denoted as Crime Graph, with vertex set $V$ of order $n = |V|$ representing (segments of) street connections and edge set $E = E^1 \cup E^2$ of size $m = |E|$ representing (segments of) streets, where $E^1$ and $E^2$ represent the set of one-way and two-way streets, respectively. In Figure 2 the topology of a city is represented as a graph where each edge represents a street segment (of variable size in meters) and each of the nine vertices the connection between these segments.

Moreover, $Q = \{Q_1, Q_2, ..., Q_q\}$ is a partition of $V$ into $q$ components (zones), *i.e.*, $Q_1 \cup Q_2 \cup ... \cup Q_q = V$ and

$Q_u \cap Q_o = \emptyset$, for every $u, o = 1...q$ with $u \neq o$. In Figure 2, three zones are defined. We also denote by $Q[i]$ the index of the component of vertex $i \in V$: *i.e.*, $i \in Q_{Q[i]}$.

Furthermore, for each edge $\{i, j\} \in E$, we define $cf_{ij}$ and $l_{ij}$ as the crime index and length of the edge, respectively. In Figure 2 we present an example of a crime graph with nine vertices and three zones, where one-way street segments are represented as single lines, two-way street segments are represented as double lines, and edge labels $(cf, l)$ represent the crime index and the length of the edge, respectively. The crime index defined for each edge is defined as the amount of crime events that happened in that street segment.

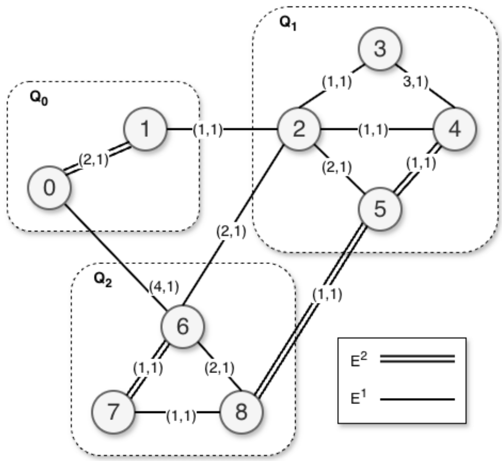


**Figure 2.** Representation of a crime graph composed of nine vertices and 13 edges, where each edge corresponds to a street segment within the urban network. Four edges are represented as bidirectional edges, reflecting streets that permit two-way traffic flow.

## 4.2 Mathematical Formulation

To present the mathematical formulation of the `RVP-Urb` problem, we introduce the following additional notation. Given the input crime graph $G^c = (V, E, Q)$, we build a directed graph $\overrightarrow{G} = (V, A, Q)$ where, for each edge $\{i, j\} \in E^2$, two arcs $(i, j)$ and $(j, i)$ are created in $A$. Similarly, for each edge $\{i, j\} \in E^1$, a single arc $(i, j)$ is added to $A$ representing its direction. The *out-neighborhood* of $i \in V$, denoted by $N^+(i)$, is the set of all vertices $j$ adjacent from $i$ in $\overrightarrow{G}$; that is, $N^+(i) = \{j|(i, j) \in A\}$. Furthermore, the *in-neighborhood* of $i \in V$, denoted by $N^-(i)$, is the set of all vertices $j$ adjacent to $i$ in $\overrightarrow{G}$; that is, $N^-(i) = \{j|(j, i) \in A\}$. We also define $N_Q^+(i) = \{j \in N^+(i) : Q[i] = Q[j]\}$ as the *zone-out-neighborhood* of a vertex $i \in V$ (the vertices in the out-neighborhood of $i$ that are in the same zone) and $N_Q^-(i) = \{j \in N^-(i) : Q[i] = Q[j]\}$ as the *zone-in-neighborhood* of vertex $i \in V$ (the vertices in the in-neighborhood of $i$ that are in the same zone). Given a subset of vertices $S \subseteq V$, we denote by $A[S]$ the subset of arcs induced in $\overrightarrow{G} = (V, A, Q)$ by $S$. Thus, we define $A_Q = A[Q_1] \cup A[Q_2] \cup ... \cup A[Q_q]$ as the set of *intra-zone* arcs, where both end points in the same zone (component).

In Figure 3(a), we illustrate the directed graph derived from the graph presented in Figure 2. In contrast, Figure 3(b) depicts a solution example of the problem, with a graph route (inter-zone) $[1 \to 0 \to 6 \to 7 \to 6 \to 2 \to 4 \to 3 \to 2 \to 1]$ (in blue) going through zones $Q_0$, $Q_1$ and $Q_2$, an intra-zone route $[4 \to 5 \to 4]$ (in red) in zone $Q_1$, and a non-drivable unit in vertex 8 (in green). Figure 3(c) represents the edges covered in the crime graph by the solution presented in Figure 3(b), with the graph route, the intra-zone route, and the non-drivable unit.Note that all edges with a non-drivable unit at one of their vertices are considered covered. Moreover, a route does not need to use both arcs of a two-way edge to cover it.

The `RVP-Urb` problem can be formulated as a mixed integer programming problem, as follows. Table 1 summarizes the set of parameters used in the proposed mathematical formulation, while Table 2 describes the binary and continuous decision variables of the model. This way, the proposed objective function (1) aims to maximize the crime index of the edges covered by the drivable and non-drivable units.

$$\max \sum_{\{i,j\} \in E} cf_{ij} \cdot c_{ij} \tag{1}$$

The following constraints (2-8) are used to construct the graph routes of the drivable units (*i.e.*, police vehicles) $K$. Constraints (2) impose the degree balance for each vertex, *i.e.*, if a unit enters a location (vertex), it must leave this location. Moreover, Constraints (3) rule that only one vertex can play the role of the start vertex in a graph route. Constraints (4-6) eliminate sub-routes by creating a positive flow through the route. The constant $M$ should be large enough to render constraints (4-5) and (6) redundant when $y_{ik} = 1$ and $x_{ijk} > 0$, respectively. Based on previous experiences, we set $M = |A|$ when CPLEX 12.4 is used to solve the mathematical formulation. The method for sub-route elimination is the one proposed by M. Ball and Bodin [1983].

$$\sum_{j \in N^+(i)} x_{ijk} = \sum_{j \in N^-(i)} x_{jik}, \qquad \forall k \in K, \forall i \in V \tag{2}$$

$$\sum_{i \in V} y_{ik} = 1, \qquad \forall k \in K, \tag{3}$$

$$\sum_{j \in N^+(i)} f_{ijk} - \sum_{j \in N^-(i)} f_{jik} \geq \sum_{j \in N^+(i)} x_{ijk} - M \cdot y_{ik}, \qquad \forall k \in K, \forall i \in V, \tag{4}$$

$$\sum_{j \in N^+(i)} f_{ijk} - \sum_{j \in N^-(i)} f_{jik} \leq \sum_{j \in N^+(i)} x_{ijk} + M \cdot y_{ik}, \qquad \forall k \in K, \forall i \in V, \tag{5}$$

$$f_{ijk} \leq M \cdot x_{ijk}, \qquad \forall k \in K, \forall (i, j) \in A \tag{6}$$

We define that every drivable unit $k \in K$ in a graph route has a set of priority edges ($E_k \subseteq E$) that must be covered (*e.g.*, schools, shopping, *etc.*) by its course. This set of mandatory street segments is imposed by Constraints (7-8).

$$x_{ijk} \geq 1, \qquad \forall k \in K, \forall \{i, j\} \in (E_k \cap E^1) \tag{7}$$

$$x_{ijk} + x_{jik} \geq 1, \qquad \forall k \in K, \forall \{i, j\} \in (E_k \cap E^2) \tag{8}$$

Constraints (9-13) are the same as (2-6) but for the intra-zone route case.

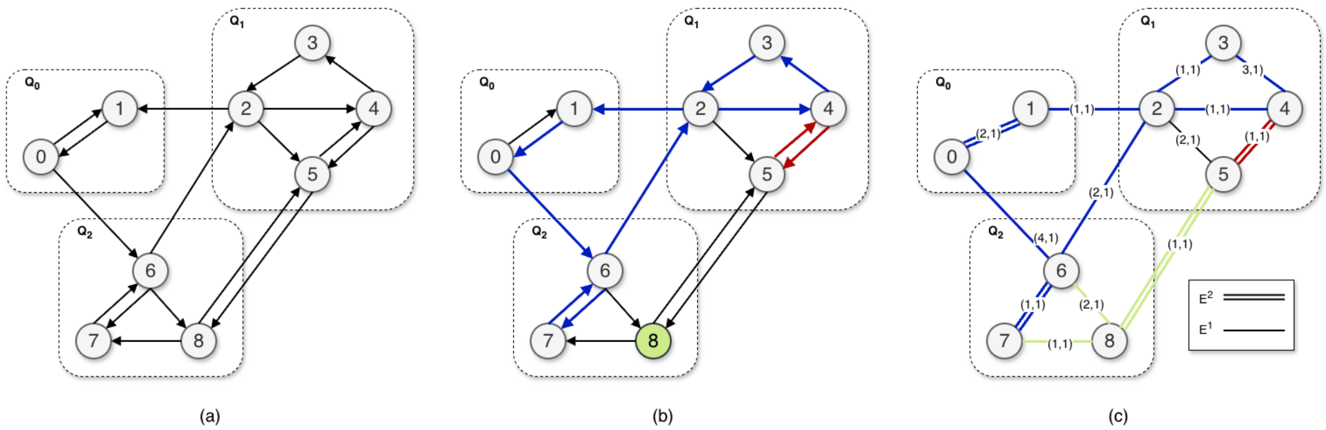


**Figure 3.** (a) Directed graph (b) Solution example (c) Covered edges in crime graph

**Table 1.** List of Data and Sets.

| *Data and Sets* | *Description* |
|---|---|
| $P$ | Set of non-drivable units (police officers). |
| $K, K_u$ | Set of drivable units (police vehicles) used in graph routes ($K$) and intra-zone routes ($K_u$) in zone $u = 1...q$. Furthermore, every drivable unit $k \in K$ in a graph route has a set of priority edges $E_k \subseteq E$ that must be covered. |
| $L_{max}, L^u_{max}$ | Maximum length of graph routes ($L_{max}$) and intra-zone routes ($L^u_{max}$) in zone $u = 1...q$. |
| $cf_{ij}$ | crime factor of edge $\{i,j\} \in E$ |
| $l_{ij}$ | length of edge $\{i,j\} \in E$ |

**Table 2.** List of variables and their descriptions.

| *Variables* | *Description* |
|---|---|
| $x_{ijk}$ | Integer variable which indicates the number of times arc $(i,j) \in A$ is traveled by drivable unit $k \in K$, in a graph route. |
| $y_{ik}$ | Binary variable which indicates whether vertex $i \in V$ is the start vertex of the graph route assign to drivable unit $k \in K$. |
| $f_{ijk}$ | Continuous variable which indicates the flow value of arc $(i,j) \in A$ in graph route $k \in K$. |
| $x^u_{ijk}$ | Integer variable which indicates the number of times arc $(i,j) \in A[Q_u]$ is traveled by drivable unit $k \in K_u$, in a intra-zone route in zone $u = 1...q$. |
| $y^u_{ik}$ | Binary variable which indicates whether vertex $i \in Q_u$ is the start vertex of the intra-zone route assign to drivable unit $k \in K_u$. |
| $f^u_{ijk}$ | Continuous variable which indicates the flow value of arc $(i,j) \in A$ in intra-route $k \in K_u$, in zone $u = 1...q$. |
| $w_i$ | Binary variable which indicates whether a non-drivable unit is allocated to guard vertex $i \in V$ or not. |
| $c_{ij}$ | Binary variable which indicates whether edge $\{i,j\} \in E$ is covered (protected) by at least one (drivable or non-drivable) unit. |

$$\sum_{j \in N^+_Q(i)} x^u_{ijk} = \sum_{j \in N^-_Q(i)} x^u_{jik}, \quad \forall u = 1...q, \forall k \in K_u, \forall i \in Q_u \tag{9}$$

$$\sum_{i \in Q_u} y^u_{ik} = 1, \quad \forall u = 1...q, \forall k \in K_u, \tag{10}$$

$$\sum_{j \in N^+_Q(i)} f^u_{ijk} - \sum_{j \in N^-_Q(i)} f^u_{jik} \geq \sum_{j \in N^+_Q(i)} x^u_{ijk} - M \cdot y^u_{ik}, \quad \forall u = 1...q, \forall k \in K_u, \forall i \in Q_u \tag{11}$$

$$\sum_{j \in N_Q^+(i)} f_{ijk}^u - \sum_{j \in N_Q^-(i)} f_{jik}^u \leq \sum_{j \in N_Q^+(i)} x_{ijk}^u + M \cdot y_{ik}^u, \qquad \forall u = 1...q, \forall k \in K_u, \forall i \in Q_u \tag{12}$$

$$f_{ijk}^u \leq M \cdot x_{ijk}^u, \qquad \forall u = 1...q, \forall k \in K_u, \forall (i,j) \in A[Q_u] \tag{13}$$

Constraints (14) and (15) impose a limit to the size (distance) of every graph route and intra-zone route, respectively, while Constraints (16) bound the number of non-drivable units (police officers) in the graph.

$$\sum_{(i,j) \in A} l_{ij} \cdot x_{ijk} \leq L_{max}, \qquad \forall k \in K \tag{14}$$

$$\sum_{(i,j) \in A[Q_u]} l_{ij} \cdot x_{ijk}^u \leq L_{max}^u, \quad \forall u = 1...q, \forall k \in K_u \tag{15}$$

$$\sum_{i \in V} w_i \leq |P| \tag{16}$$

Inequalities (17-20) link the drivable and non-drivable resources with the cover variables $c$ to establish which edges are covered (protected by a police officer or a police vehicle). An outside edge (*i.e.*, an edge with each vertex in a different zone) is covered if its corresponding arc(s) belong(s) to a route or at least one endpoint has a non-drivable unit (17-18). Similarly, a zone edge (*i.e.*, an edge with both vertices in the same zone) is covered if its corresponding arc(s) belong(s) to a route, an intra-zone route, or at least one vertex has a non-drivable unit (19-20).

$$c_{ij} \leq \sum_{k \in K} x_{ijk} + w_i + w_j, \qquad \forall \{i,j\} \in E^1, \text{ such } Q[i] \neq Q[j] \tag{17}$$

$$c_{ij} \leq \sum_{k \in K} (x_{ijk} + x_{jik}) + w_i + w_j, \qquad \forall \{i,j\} \in E^2, \text{ such } Q[i] \neq Q[j] \tag{18}$$

$$c_{ij} \leq \sum_{k \in K} x_{ijk} + \sum_{k \in K_u} x_{ijk}^{Q[i]} + w_i + w_j, \quad \forall \{i,j\} \in E^1, \text{ such } Q[i] = Q[j] \tag{19}$$

$$c_{ij} \leq \sum_{k \in K} (x_{ijk} + x_{jik}) + \sum_{k \in K_u} (x_{ijk}^{Q[i]} + x_{jik}^{Q[i]}) + w_i + w_j, \qquad \forall \{i,j\} \in E^2, \text{ such } Q[i] = Q[j] \tag{20}$$

Finally, constraints (21) and (22) are not necessary for the model, but their inclusion helps to eliminate symmetric solutions. These constraints break the symmetry of the routes by ordering them by total route length.

$$\sum_{(i,j) \in A} l_{ij} \cdot x_{ijk} \leq \sum_{(i,j) \in A} l_{ij} \cdot x_{ij(k+1)}, \quad \forall k = 1...|K| - 1 \tag{21}$$

$$\sum_{(i,j) \in A[Q_u]} l_{ij} \cdot x_{ijk}^u \leq \sum_{(i,j) \in A[Q_u]} l_{ij} \cdot x_{ij(k+1)}^u, \quad \forall u = 1...q, \forall k = 1...|K_u| - 1 \tag{22}$$

# 5 PROTECTOR: an Heuristic for the RVP-Urb Problem

Defining the preventive patrol routes for several drivable units can be a complex task, depending on some factors, *e.g.*, the size of the crime graph, the number of hotspots, *etc.* The number of possible routes for each drivable unit may grow exponentially depending on the previously mentioned factors, making the adoption of metaheuristic-based heuristics a natural solution for the RVP-Urb problem, since exact solutions do not scale. In this section, we introduce PROTECTOR, the hybrid heuristic based on the combination of the GRASP metaheuristic [Resende and Ribeiro, 2016], with Variable Neighborhood Descent (VND) local search [Duarte *et al.*, 2018], combining local improvement procedures with higher-level approaches to escape local minima and conduct a thorough search across the solution space, ensuring robust exploration, which is desirable for the RVP-Urb problem.

GRASP is an iterative metaheuristic that has been applied to a wide range of optimization problems [Resende and Ribeiro, 2016]. Each iteration of the GRASP procedure consists of two different and complementary phases. In the first phase, known as the *construction phase*, a feasible solution, *i.e.*, a viable preventive patrol route, is generated in a greedy but randomized manner. In the subsequent *local search phase*, the algorithm explores the neighborhood of the constructed solution, *i.e.*, alternative route configurations derived from the initial one. This exploration aims to iteratively refine the current solution by reducing its cost and ultimately discovering higher-performing routes. Among all iterations, the best solution, corresponding to the route that covers as many hotspots as possible, is generated as the final outcome.

In the context of PROTECTOR, the first phase corresponds to the construction phase. It is composed of three main algorithms designed to generate candidate patrol routes: (i) the generation of graph routes that prioritize specific edges, (ii) the generation of graph routes without such priorities, and (iii) the creation of intra-zone routes within defined territorial boundaries. Once these routes are constructed, the method concludes by adding all remaining non-drivable units into the final solution, ensuring a comprehensive and operationally feasible preventive patrol plan.

The method presented in Algorithm 1 receives as input the crime graph $G^c$ and a drivable unit $k$, and builds a preventive patrol route with priority edges (the ones that are mandatory to be covered) for this unit, aiming to connect them into a closed path. First, let $\mathbb{P}(E_k)$ denote the family of all permutations of the priority edges set $E_k$, for each route $k \in K$ in the crime graph $G^c$. In practice, this set is very small, with an average of approximately 3 priority edges per route and a maximum of 10 in our test instances, which keeps the enumeration computationally tractable. Initially, a loop that explores all permutations $P_E \in \mathbb{P}(E_k)$ is executed (line 1). In each iteration of this loop, a route $R_k$ is initialized with the first priority edge of the set (line 2). Then the method traverses all remaining priority edges that still need to be covered, using a minimum-path algorithm for each to find a viable path from the partial route $R_k$ to the considered priority edge (lines 5-8). Priority edges that are incidentally covered by a minimum path are dynamically removed from $P_E$, preventing redundant processing. In line 9, the route is closed by connecting the last vertex to the first vertex. If the length of the route is feasible (*i.e.*, respects the distance limit $L_{max}$ define by the police officer), the algorithm finishes and returns the constructed route (lines 10-12). Otherwise, another permutation is considered until a feasible route is generated. Parameter $L_{max}$ acts as an operational constraint that prevents the generation of excessively long routes and rarely invalidates reasonable routes, which prevents the exhaustive exploration of all permutations. In the current version of `PROTECTOR`, we use Dijkstra's algorithm [Cormen *et al.*, 2001] to find the minimum path from an origin to a destination.

**Algorithm 1** `Priority_Graph_Route_Builder`($G^c$,$k$)

1: **for** $P_E \in \mathbb{P}(E_k)$ **do**
2: $\quad R_k \leftarrow \overrightarrow{ij} \leftarrow$ `Get_First_Edge`($P_E$) ▷ First route edge
3: $\quad tail \leftarrow j$
4: $\quad P_E \leftarrow P_E \backslash \{\overrightarrow{ij}\}$
5: $\quad$ **for** $\overrightarrow{uv} \in P_E$ **do**
6: $\quad\quad R_k \leftarrow R_k \cup$ `Min_Path`($G^c$,`tail`,`u`) $\cup\, \overrightarrow{uv}$
7: $\quad\quad tail \leftarrow v$
8: $\quad$ **end for**
9: $\quad R_k \leftarrow R_k \cup$ `Min_Path`($G^c$,`tail`,`i`) ▷ Last route edge
10: $\quad$ **if** `Distance` $(R_k) \leq L_{max}$ **then**
11: $\quad\quad$ **return** $R_k$
12: $\quad$ **end if**
13: **end for**

The generation of graph routes without considering priority edges is presented in Algorithm 2. First, a candidate list (*CL*) is built by considering the $\alpha \times |V|$ vertices in $G^c$ with the highest crime indexes (line 1). As aforementioned, we define the crime index of each vertex as the sum of the crime events over a specific time interval for each edge connected to that vertex. Note that parameter $\alpha \in (0, 1]$ controls the proportion of vertices considered during the construction of the candidate list. Smaller values of $\alpha$ lead to a more greedy selection focused on the highest-crime vertices, while values closer to 1 promote a more diversified and less greedy choice. A vertex $i \in CL$ is randomly selected (line 3), together with its neighbor $j \in N^+(i)$ with the highest edge crime index (line 4). The route $R_k$ is then defined as the arc $\overrightarrow{ij}$ and the minimum path connecting vertices $j$ and $i$ (line 5). Please recall that if the route length is feasible, the algorithm terminates and returns the generated route (lines 6 and 7). Otherwise, the vertex is discarded, and another one is considered until a feasible route is generated. It is important to note that the method for generating intra-zone routes is the same as that for graph routes without priority edges, but with the restriction that it applies solely within a specific zone $u = 1...q$. In addition to the generated routes, each non-drivable unit is randomly placed at a graph vertex to create an initial solution.

**Algorithm 2** `Graph_Route_Builder`($G^c, \alpha, k$)

1: $CL \leftarrow$ `Candidates(V,`$\alpha$`)`
2: **while** $CL \neq \emptyset$ **do**
3: $\quad i \leftarrow$ `Get_Candidate(CL)`
4: $\quad j \leftarrow$ `Highest_Crime_Neig(`$G^c$`,i)`
5: $\quad R_k \leftarrow \overrightarrow{ij} \cup$ `Min_Path(`$G^c$`, j, i)`
6: $\quad$ **if** `Distance`$(R_k) \leq L_{max}$ **then**
7: $\quad\quad$ **return** $R_k$
8: $\quad$ **else**
9: $\quad\quad CL \leftarrow CL \backslash \{i\}$
10: $\quad$ **end if**
11: **end while**

Once the initial solution is generated, *i.e.*, a preventive patrol route, the GRASP method initiates a series of local search movements to improve the current solution. This intensification phase includes three neighborhood components, evaluated using VND [Duarte *et al.*, 2018], which is applied in each neighborhood in a deterministic manner. Upon discovering an improvement, the method triggers a restart in the local search, restarting the VND process from the initial neighborhood. In `PROTECTOR`, the local search considers each route belonging to the current solution separately. In this way, movements are successively applied to the route until improvement is no longer possible, and only then they move on to the following generated route. The VND procedure concludes when no neighborhood offers further improvements. Next, we will introduce possible search movements to improve the current solution, *i.e.*, the preventive patrol route.

**Path Switch:** This movement consists of exploring the neighborhood of route $R$ by considering the removal of a continuous set of edges from the route (excluding priority edges) and reconnecting them while maintaining the maximum route length constraint. First, the method tries to remove all paths with size equal to 1, but if no improvement is found, the size of the removed path is iteratively increased by 1 until the length of the removed path reaches the limit value of (`max_section` $\times$ `Distance(R)`). The parameter `max_section` defines the maximum relative size of the route segment that can be removed during the path removal operation, expressed as a fraction of the total route $R$ length . The route reconnection process is based on finding the maximum spanning tree using a variation of Prim's Algorithm [Cormen *et al.*, 2001]. For each removed path $(v_1, ..., v_p)$, the method starts its spanning tree from vertex $v_1$, and then adds edges (along with vertices) one by one to the tree, selecting edges with greater crime index until vertex $v_p$ is reached. Figure 4 illustrates this move.

**Loop Insertion:** The formulation presented in Section 4.2 defines routes as closed walks (*i.e.*, both vertices and edges can

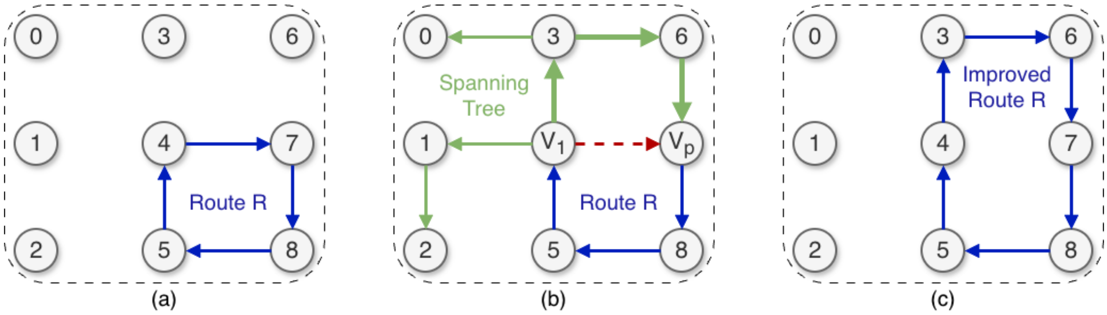


**Figure 4.** The path switch neighborhood. The route $R = [4 \to 7 \to 8 \to 5 \to 4]$ is expanded to $R = [4 \to 3 \to 6 \to 7 \to 8 \to 5 \to 4]$.

be repeated by the drivable unit). Therefore, we developed the loop addition movement, where again we considered all paths of vertices in route $R$, starting from a single edge and iteratively increasing its size (by 1), until the path's length hits the threshold of ($\texttt{max\_loop} \times \texttt{Distance(R)}$). For each path $(v_1, \ldots, v_p)$ considered, the method creates a new route connecting $v_p$ to $v_1$. As a result, the original route can be extended to include this new segment (without removing the original path), encompassing additional edges and ultimately enhancing the crime coverage of the present solution. Similarly to `max_section`, the parameter `max_loop` limits the maximum relative size of segments that can be extended, expressed as a fraction of the total route $R$ length. Moreover, the new path $(v_p, ..., v_1)$ is generated by the same process as the previous neighborhood, based on finding the maximum spanning tree from vertex $v_p$, until vertex $v_1$ is reached. Figure 5 illustrates this move.

**Guard Shift:** the non-drivable units' movement shifts each unit to a more favorable position based on the crime index of each vertex. It selects the graph's vertex with the highest available crime index for each fixed unit and relocates the unit accordingly. As mentioned earlier, a vertex's crime index is defined as the total number of crime events on its edges within a specific time interval that have not yet been covered.

Algorithm 3 illustrates the proposed GRASP-based `PROTECTOR` heuristic. Each iteration begins by constructing an initial solution (lines 3-13) and then proceeds with a local search method (line 14). The update for the best overall solution occurs within lines 15-17. This entire sequence repeats until the stopping criterion is met, specifically, until the total number of GRASP iterations $MaxIT$ is reached. In our experiments, this maximum number of iterations is set to $MaxIt = 100$. This value was determined based on preliminary experiments, which showed that 100 iterations were sufficient to achieve stable and competitive solutions for both small and large instances, without incurring unnecessary computational effort. The source code of `PROTECTOR` will be publicly available in our research group's GitHub repository at `https://github.com/UFFeScience/protector`.

# 6 Experimental Evaluation

This section evaluates the effectiveness and performance of `PROTECTOR` on real-world data for generating preventive police patrol routes. We begin by describing the dataset used to produce the instances evaluated. Next, we outline the computational environment used in the experiments, followed by a detailed case study illustrating the step-by-step application of `PROTECTOR`. We evaluate the proposed method's results against exact solutions, analyze its performance on large-scale instances, and finally perform a sensitivity analysis.

## 6.1 Crime Dataset

All data used in this experimental evaluation were obtained from the `PolRoute-DS` dataset [Cunha Sá *et al.*, 2022; Sá *et al.*, 2021], which compiles official crime records provided by the São Paulo State Police (SSP-SP)[6]. Although the dataset was downloaded in August 2019, it includes crime incidents recorded over a 15-year period, spanning from 2003 to 2019. `PolRoute-DS` encompasses multiple categories of criminal activity, namely: (i) Femicide, (ii) Mobile Phone Theft, (iii) Mobile Phone Robbery, (iv) Murder, (v) Robbery, (vi) Vehicle Theft, and (vii) Vehicle Robbery. For this evaluation, the categories femicide and murder were excluded, as these crimes are typically not preventable through proactive patrol routing strategies.

The `PolRoute-DS` dataset is implemented as a Data Warehouse [Inmon *et al.*, 2005] modeled according to the snowflake schema. This multidimensional structure comprises fact and dimension tables that support analytical processing and spatiotemporal aggregation. Fact tables store quantitative indicators, *e.g.*, crime index, while dimension tables provide the contextual attributes necessary to characterize these measures. The current schema comprises seven tables: (i) *Crime*, (ii) *Segment*, (iii) *Vertex*, (iv) *Time*, (v) *District*, (vi) *Zone*, and (vii) *Neighborhood*. The *Crime* table records the number of incidents for each crime category, associating them with specific map segments and time periods. This table can also store pre-aggregated data for extended time periods, enhancing query performance. The *Time* table defines the temporal hierarchy of crime events across multiple granularities (*e.g.*, hour, day, month, year).

The *Segment* table stores the street segments that compose the crime graph, while the *Vertex* table contains the corresponding vertices, representing intersections or other relevant urban points. The *Zone* table specifies user-defined areas on the map that can be employed to generate patrol routes. The *District* table represents administrative divisions within the city, and the *Neighborhood* table lists officially recognized neighborhoods. Table 3 reports the total number of tuples

[6] `http://www.ssp.sp.gov.br/transparenciassp/Consulta.aspx`

**Algorithm 3** `PROTECTOR`($G^c, MaxIT, \alpha$, `max_section`, `max_loop`)

$S^* \leftarrow \emptyset$
**for** $i = 1$ to $MaxIT$ **do**
  $S \leftarrow \emptyset$
  **for** $k \in K$ **do**
    **if** $E_k \neq \emptyset$ **then**
      $S \leftarrow S \cup$ `Priority_Graph_Route_Builder`($G^c, k$)
    **else**
      $S \leftarrow S \cup$ `Graph_Route_Builder`($G^c, \alpha, k$)
    **end if**
  **end for**
  **for** $u = 1...q$ **and** $k \in K_u$ **do**
    $S \leftarrow S \cup$ `Intra-Zone_Route_Builder`($G^c, \alpha, u, k$) ▷ `Graph_Route_Builder`($G^c, \alpha, k$) but for a specific zone
  **end for**
  $S \leftarrow$ `VND`($G^c$,$S$, `max_section`, `max_loop`)
  **if** `Cost`($S$) > `Cost`($S^*$) **then**
    $S^* = S$
  **end if**
**end for**
**return** $S^*$

stored in each table of the `PolRoute-DS` schema.

**Table 3.** Total number of tuples per table.

| Table | Number of Tuples |
|---|---|
| Crime | 7,267,353 |
| Time | 28,593 |
| Zone | 115 |
| Segment | 226,975 |
| Vertex | 180,598 |
| District | 97 |
| Neighborhood | 323 |

## 6.2 Environment Setup

All experiments were executed on a computer equipped with an Intel® Core™ i7-8565U CPU @ 3.70 GHz and 16 GB of RAM, running an Ubuntu-based operating system. The `PROTECTOR` heuristic was implemented in Rust using the rustc 1.63.0 compiler, while the interface for generating the crime graph was developed in Python 3.10. The mathematical formulation was solved using IBM ILOG CPLEX version 22.1, with a maximum execution time of 7,200 seconds and all remaining parameters set to their default values. This time limit is consistent with common practice in the mixed-integer programming literature and reflects realistic decision-making horizons for patrol planning. Additional experiments with longer runtimes showed that only small instances could be solved to optimality, while large and realistic instances remained unsolved, reinforcing the need for alternative solution approaches.

Each experiment involving `PROTECTOR` was performed ten times using distinct random seeds to mitigate the influence of stochastic variation on the results. The parameter configuration of the `PROTECTOR` heuristic was optimized using the irace automatic parameter tuning framework [López-Ibáñez *et al.*, 2016], based on three benchmark instances of different sizes (in terms of number of vertices), which were selected to represent different problem scales and excluded from subsequent comparative analyses to avoid bias (see Table 5). The parameter definitions, their respective search intervals, and the best-performing values identified by irace are summarized in Table 4. Moreover, all parameters were tuned as continuous variables, and the tested intervals shown in Table 4 correspond to the full variation ranges explored by irace during the tuning process.

**Table 4.** Optimal parameter values obtained after tuning with the `IRACE` package.

| Parameter | Tested Interval | Best Value |
|---|---|---|
| $\alpha$ | [0.00, 0.99] | 0.1436 |
| `max_section` | [0.10, 0.40] | 0.3719 |
| `max_loop` | [0.10, 0.40] | 0.2620 |

## 6.3 Preliminary Evaluation

To conduct a preliminary evaluation of the `PROTECTOR` heuristic, we selected the municipality of Jandira, a small city in the Greater São Paulo region of Brazil. According to the 2019 population estimation provided by IBGE[7], Jandira has an estimated population of 124,937 inhabitants. For this initial analysis, we defined a single operational zone encompassing the city's entire urban area and considered the period from April to June 2012. Within this period, a total of seven crime incidents were considered, as detailed in Table 6. Jandira was selected as the study area due to several characteristics of the city. First, the number of recorded crime incidents is relatively low, which eases analysis of the generated routes by reducing excessive noise from high-crime-density regions.

[7] `https://agenciadenoticias.ibge.gov.br`

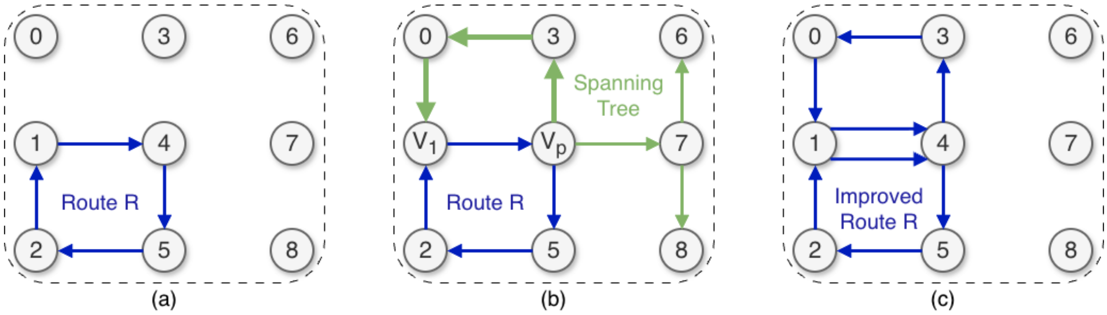


**Figure 5.** The loop insertion neighborhood. The route $R = [1 \to 4 \to 5 \to 2 \to 1]$ is expanded to $R = [1 \to 4 \to 3 \to 0 \to 1 \to 4 \to 5 \to 2 \to 1]$.

**Table 5.** Tuning instance set used in the `IRACE` package.

| **Instance** | $\lvert V\rvert$ | $\lvert E\rvert$ | $\lvert Q\rvert$ | $\lvert P\rvert$ | $\lvert K\rvert$ | $\lvert K_u\rvert$ | $L_{max}$ | $L^u_{max}$ |
|---|---|---|---|---|---|---|---|---|
| `complete-492` | 492 | 859 | 3 | 10 | 2 | 3 | 2482.41 | 772.65 |
| `vila-progresso-1988` | 1988 | 5161 | 3 | 12 | 4 | 3 | 3781.87 | 1393.32 |
| `vila-formosa-3460` | 3460 | 7923 | 5 | 13 | 5 | 2 | 6356.24 | 2341.77 |

Second, Jandira covers a very small geographic area, making it unnecessary to subdivide the city into multiple regions. This characteristic enables us to analyze routes across the entire urban area rather than focusing on selected subregions. Although defining specific regions is possible in `PROTECTOR`, and this could eliminate unused areas, reduce the search space, and improve computational efficiency, our goal was to evaluate `PROTECTOR`'s performance at the scale of a complete city.

The first step consisted of mapping all the streets in the city and classifying each of them according to their directionality, either one-way or two-way, as presented in Figure 6(a). Following, the streets were partitioned into smaller segments to facilitate a more granular spatial analysis. In this preliminary evaluation, each street was divided into segments up to 250 meters long, a parameter that can be adjusted to suit the desired level of detail. Larger segment sizes aggregate more crimes within a single edge, thereby reducing the granularity and flexibility of the resulting crime graph when generating patrol routes. After this segmentation process, a new graph was generated in which each edge and vertex was assigned an associated crime index, as shown in Figure 6(b).

Once the crime graph was constructed, the `PROTECTOR` heuristic was executed using the following parameter values: $P = 0$, $K_1 = 4$, $L_{\max} = 5000$, and $L^1_{\max} = 5000$. Under these settings, four intra-zone patrol routes were generated, as presented in Figure 7. It is clear from Figure 7 that `PROTECTOR` successfully produced preventive patrol routes that consistently cover edges associated with recorded crime events, which are represented as golden dots.

Given that `PROTECTOR` constructs routes in the form of closed walks and that the maximum route length was limited to 5 km, the algorithm occasionally produces relatively long routes that primarily cover a single crime location. However, this behavior is dependent on the density of crime events in the crime graph; if additional crime incidents were added, the generated route lengths could be adjusted accordingly.

This preliminary evaluation demonstrates that `PROTECTOR` is capable of generating patrol routes that cover all recorded crime incidents within the analyzed period. It is important to note that `PROTECTOR` prioritizes generating routes that remain as close as possible to identified crime hotspots. As a consequence, certain areas of the city may not be covered by the generated routes, as observed in the southern region of Jandira. Although in an ideal scenario a sufficient number of patrol vehicles would be available to ensure complete coverage of the entire city, in practice, the number of available vehicles is typically smaller than the number required for full urban coverage, as reported in Turchan and Braga [2024]. Therefore, `PROTECTOR` focuses on maximizing coverage of crime hotspots rather than uniformly covering the entire city. Furthermore, `PROTECTOR` is intended to run periodically, as crime hotspots may shift over time, requiring patrol routes to be continuously updated to reflect these changes. In the following sections, we present a detailed comparison between the routes generated by `PROTECTOR` and the exact solutions obtained from the mathematical model for small-scale instances. In addition, we evaluate the performance of the proposed heuristic on larger and more complex problem instances.

## 6.4 Comparing `PROTECTOR` with the Exact Solution

To compare the exact solution, *i.e.*, the optimal route obtained by CPLEX when solving the mathematical model described in Section 4.2, and the solutions produced by the `PROTECTOR` heuristic, a set of 13 small-scale instances was generated from the dataset detailed in Subsection 6.1. Each instance was carefully designed to be of manageable size, ensuring that the exact method could obtain an optimal solution within a reasonable time. This constraint was necessary because applying the exact method to medium-sized or large-sized instances would be computationally impractical. Table 7 summarizes the experimental results, comparing the solutions produced by `PROTECTOR` with those obtained from the exact mathematical formulation and a baseline greedy algorithm. The latter mimics a human-like, greedy decision-making process in which patrol units, whether drivable or non-drivable, prioritize visiting the next vertex or edge with the highest crime index.

**Table 6.** Sample of recorded crime events in the city of Jandira, covering the period from April to June 2012, including spatial coordinates, time of the event, and crime description.

| Date | Latitude | Longitude | Period | Crime Description |
|---|---|---|---|---|
| 21/04/2012 | -23.529111 | -46.903865 | Afternoon | Theft (art. 155) – Vehicle |
| 01/06/2012 | -23.529995 | -46.901177 | Afternoon | Theft (art. 155) – Vehicle |
| 01/06/2012 | -23.529995 | -46.901177 | Afternoon | Theft (art. 155) – Other Cases |
| 01/06/2012 | -23.529995 | -46.901177 | Afternoon | Theft (art. 155) – Other Cases |
| 05/06/2012 | -23.530754 | -46.913496 | Night | Theft (art. 155) – Other Cases |
| 06/06/2012 | -23.548094 | -46.898549 | Morning | Theft (art. 155) – Other Cases |
| 09/06/2012 | -23.516374 | -46.915316 | Morning | Theft (art. 155) – Other Cases |

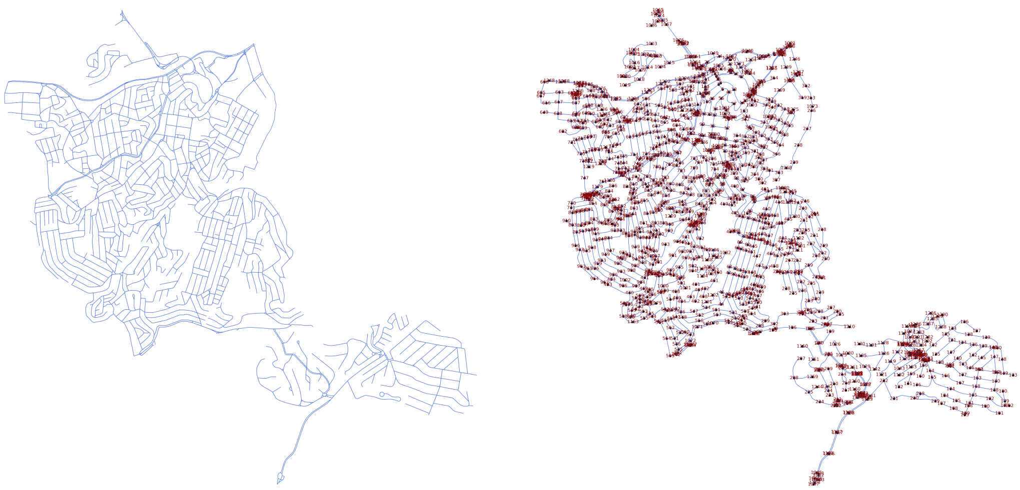

**Figure 6.** (a) The street network of the city of Jandira, São Paulo, Brazil. (b) The corresponding crime graph generated based on the city's street layout, where each vertex (in red) and edge represent the spatial structure used for preventive route generation.

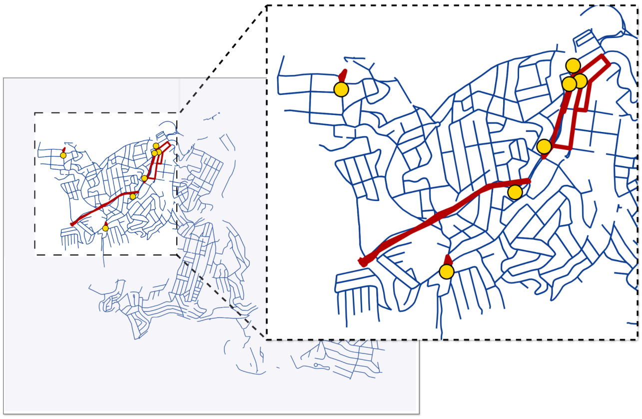

**Figure 7.** The four preventive police patrol routes generated by the PROTECTOR heuristic for the city of Jandira, São Paulo, Brazil, considering the spatial distribution of crime events summarized in Table 6.

The first column reports the name of each scenario, *i.e.*, an instance, considered in the experiments. Columns 2 and 3 present, respectively, the objective function value and the computational time of the exact method used to generate the routes. Similarly, Columns 4 and 5 display the corresponding objective function value and execution time achieved by the greedy approach. Column 6 indicates the best solution value obtained by the PROTECTOR heuristic after 10 runs. In contrast, Columns 7 and 8 summarize the average values of the objective function and execution time, respectively, over those runs. Values shown in boldface denote cases in which the corresponding method successfully reached the optimal solution. Finally, Column 9 reports the relative gap between the best solution identified by PROTECTOR and the optimal solution obtained by the exact method. This performance gap is normalized according to the following expression:

$$\mathrm{gap}(\%) = \frac{\mathtt{PROTECTOR} - \mathit{Exact}}{\mathit{Exact}} \times 100 \tag{23}$$

The results show that the proposed mathematical model obtained proven optimal solutions for all small instances, which were intentionally designed to have at most 75 vertices. Although instances mini1-64 and mini2-75 required much longer computational times, they were still solved to optimality within the imposed time limit. On the other hand, the greedy heuristic achieved fast execution times. Still, it yielded lower solution quality, with an average objective function value of 207.7, compared to 282.28 (the higher, the better) obtained by the exact method. The PROTECTOR heuristic, on the other hand, produced high-quality solutions across the same set of instances, attaining optimality in 7 of them (or in 4 cases when considering only the average objective function value). For all instances, PROTECTOR achieved a gap of up to 6.16% with respect to the optimal solutions provided by the exact method (1.21% in average). It is also worth noting that the execution times recorded for PROTECTOR were substantially shorter than those required by CPLEX, averaging approximately 0.03 seconds, particularly in instances mini1-64 and mini2-75, where the exact method faced computational difficulties.

## 6.5 Evaluation with Large-Scale Instances

We conducted two experimental studies to evaluate the effectiveness of PROTECTOR in generating preventive patrol routes for large-scale instances of the problem. The first experiment focuses on a single case study corresponding to the Sacomã district in São Paulo (sacoma-2480). In this experiment, the patrol routes produced by PROTECTOR are compared against two benchmarks: (i) the routes manually designed by domain experts from the Police Department, and (ii) the routes obtained using the greedy heuristic approach. It is important to note that the solutions developed by police experts are extremely time-consuming and labor-intensive. Consequently, the authors were able to obtain expert-designed routes for this single instance only. Furthermore, as no prior research has examined the RVP-Urb problem, there are no other available expert or benchmark solutions for comparison. Therefore, in the second experiment, which involves the remaining large-scale instances, the performance of PROTECTOR is evaluated solely relative to the greedy heuristic.

Table 8 presents the results obtained from the first experiment involving domain experts. The first column reports the instance name, while the second column presents the corresponding solution value, *i.e.*, the objective function value defined in Section 4.2, computed based on the routes designed by the police experts. Columns three through eight provide analogous information to columns four through nine of Table 7. In this case, however, the column labeled gap represents the difference between the best solution identified by PROTECTOR and that obtained by the domain experts. The solution designed by the police experts achieved an objective value of $10{,}412.70$. On the other hand, the greedy heuristic, while computationally efficient, produced an inferior solution valued at only $6{,}520.52$. This discrepancy indicates that the expert-generated solution is not merely the outcome of a straightforward greedy process; rather, it reflects the influence of professional judgment and experience.

Remarkably, PROTECTOR was able to generate a superior solution, achieving an objective value of $14{,}834.42$ in only 33 seconds. This represents a $42.46\%$ improvement over the solution produced by the experts. The superior performance of PROTECTOR can be attributed to its algorithmic ability to explore a wide range of possible patrol configurations. In contrast, human cognitive limitations and reliance on established heuristics limit the effectiveness of expert-designed solutions. In practice, police experts tend to replicate patrol routes that have proven "successful" in the past, thereby restricting the exploration of alternative or potentially more effective strategies. Furthermore, manually designing these routes is highly time-consuming and labor-intensive, often requiring extensive deliberation and iterative discussions that occur on an almost daily basis.

As previously discussed, additional solutions produced by police experts were not available and could not be analyzed. Consequently, in our second experiment involving large-scale instances, we compare the performance of PROTECTOR exclusively against the greedy heuristic. The exact mathematical model was also tested on these large instances; however, CPLEX could not be used due to memory limitations, preventing the generation of any feasible solution within the

**Table 7.** Comparison of the routes generated by PROTECTOR with the one generated exact solution

| Instance | Exact | Time (s) | Greedy | Time (s) | PROTECTOR Best | PROTECTOR Avg. | Time (s) | Gap (%) |
|---|---|---|---|---|---|---|---|---|
| cjn1-18 | **126.74** | 0.15 | 118.35 | $\leq$ 0.01 | 125.74 | 124.70 | $\leq$ 0.01 | -0.79 |
| cjn2-18 | **136.91** | 0.16 | 114.65 | $\leq$ 0.01 | 134.60 | 134.60 | $\leq$ 0.01 | -1.69 |
| cjn1-24 | **171.91** | 0.44 | 138.40 | $\leq$ 0.01 | **171.91** | 170.82 | $\leq$ 0.01 | 0.00 |
| cjn2-25 | **231.86** | 0.36 | 178.95 | $\leq$ 0.01 | **231.86** | 228.48 | 0.01 | 0.00 |
| cjn1-33 | **270.96** | 0.35 | 226.35 | $\leq$ 0.01 | **270.96** | 270.16 | 0.04 | 0.00 |
| cjn2-35 | **305.36** | 0.11 | 245.74 | $\leq$ 0.01 | **305.36** | **305.36** | 0.04 | 0.00 |
| cjn1-40 | **335.52** | 1.53 | 212.63 | $\leq$ 0.01 | **335.52** | **335.52** | 0.04 | 0.00 |
| cjn2-41 | **360.36** | 0.21 | 286.86 | $\leq$ 0.01 | **360.36** | **360.36** | 0.04 | 0.00 |
| cjn1-46 | **401.14** | 0.49 | 318.59 | $\leq$ 0.01 | **401.14** | **401.14** | 0.11 | 0.00 |
| cjn2-50 | **414.41** | 0.27 | 298.92 | $\leq$ 0.01 | 411.41 | 411.41 | 0.06 | -0.72 |
| mini1-64 | **403.65** | 3,606.32 | 273.90 | $\leq$ 0.01 | 379.43 | 370.49 | 0.03 | -6.00 |
| mini2-75 | **403.66** | 594.46 | 206.59 | $\leq$ 0.01 | 378.80 | 378.80 | 0.03 | -6.16 |
| mini3-20 | **108.48** | 0.14 | 80.21 | $\leq$ 0.01 | 108.08 | 108.08 | $\leq$ 0.01 | -0.37 |
| **Avg.** | 282.38 | 323.46 | 207.70 | 0.00 | 278.09 | 276.91 | 0.03 | -1.21 |

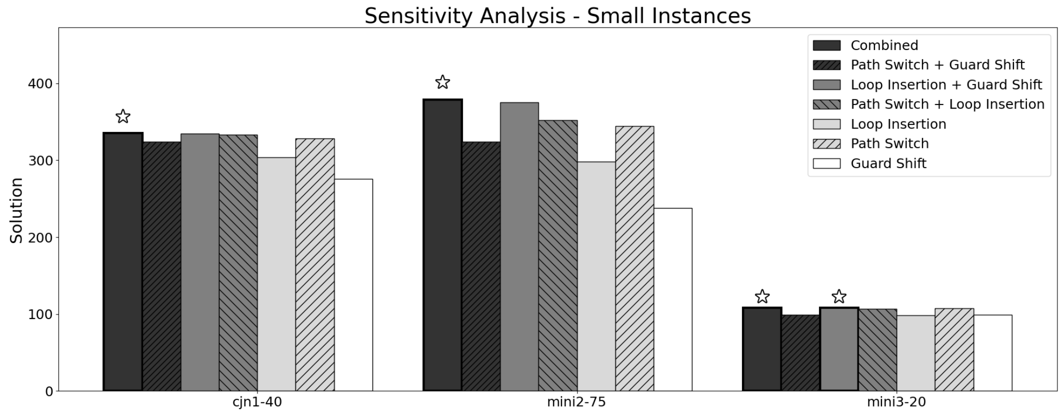


**Figure 8.** sensitivity Analysis of PROTECTOR for small-scale instances.

available computational resources. Table 9 presents the results of this comparison for the remaining 21 large instances. The columns in this table follow the same structure and meaning as those in the previous tables; however, in this case, the gap column represents the relative difference between the best solution obtained by PROTECTOR and that produced by the greedy approach.

The results clearly indicate that PROTECTOR consistently outperforms the greedy heuristic across all instances. Specifically, the greedy approach achieved an average solution value of only 2,685.93, whereas PROTECTOR presented a higher average value of 7,671.73, with an average computational time of 27.30 seconds. Although the greedy heuristic is computationally very efficient, its solutions are of considerably lower quality and less reliable than those generated by PROTECTOR. This is further evidenced by the large average gap of 170.85%, demonstrating PROTECTOR's superiority in producing high-quality solutions for large-scale instances.

## 6.6 Sensitivity Analysis

To understand the contribution of each neighborhood structure used in the PROTECTOR heuristic, we performed a sensitivity analysis evaluating the impact of the three local search movements employed: (i) Path Switch, (ii) Loop Insertion, and (iii) Guard Shift. The experiments were conducted by activating each movement individually, in pairwise combinations, and in the full configuration comprising all three movements. The objective value corresponds to the total crime coverage achieved by the generated patrol solution. The results are presented separately for small (Figure 8) and large (Figure 9) instances, allowing us to evaluate the behavior of the neighborhood structures across different problem scales.

Figure 8 presents the results for the small instances cjn1-40, mini2-75, and mini3-20. In all cases, the configuration that combines the three movements (Path Switch + Loop Insertion + Guard Shift) provides the best solution, demonstrating that the interaction between route modifications and the repositioning of non-drivable units offers advantages even in smaller instances. Across pairwise combinations, performance varies slightly from instance to instance. For cjn1-40, the combinations Loop Insertion + Guard Shift and Path Switch + Loop Insertion achieve solutions close to the combined configuration. A similar trend is observed in mini2-75, where Loop Insertion + Guard Shift yields results very close to the best solution. For mini3-20, all pairwise combinations perform relatively similarly, suggesting that the search space for this instance is more limited. When analyzing individual movements, Loop Insertion and Path Switch generally outperform Guard Shift, which consistently yields the lowest

**Table 8.** Evaluation with domain experts of the routes generated by PROTECTOR for the sacoma-2480 instance.

| Instance | Experts | Greedy | Time (s) | PROTECTOR Best | PROTECTOR Avg. | Time (s) | Gap (%) |
|---|---|---|---|---|---|---|---|
| sacoma-2480 | 10412.70 | 6520.52 | $\leq 0.01$ | 14834.42 | 14564.54 | 33.88 | 42.46 |

**Table 9.** Results for large instances.

| Instance | Greedy | Time (s) | PROTECTOR Best | PROTECTOR Avg. | Time (s) | Gap (%) |
|---|---|---|---|---|---|---|
| parque-santa-madalena-3578 | 4,407.92 | $\leq 0.01$ | 12,482.05 | 12,277.96 | 48.69 | 183.17 |
| vila-matilde-3454 | 2,812.55 | $\leq 0.01$ | 10,992.41 | 10,808.13 | 55.33 | 290.83 |
| vila-santa-teresa-3222 | 3,691.52 | $\leq 0.01$ | 12,051.71 | 11,897.20 | 42.90 | 226.47 |
| americanopolis-2897 | 3,699.46 | 0.04 | 9,071.50 | 8,844.42 | 16.42 | 145.21 |
| vila-renato-645 | 1,171.67 | $\leq 0.01$ | 1,356.43 | 1,350.82 | 0.11 | 15.76 |
| perdizes1-1793 | 3,280.45 | $\leq 0.01$ | 6,276.55 | 6,165.80 | 41.18 | 91.31 |
| vila-solange-3372 | 3,260.93 | $\leq 0.01$ | 11,957.11 | 11,691.97 | 51.30 | 266.67 |
| cidade-tiradentes-3562 | 2,640.58 | $\leq 0.01$ | 6,704.68 | 6,583.79 | 19.65 | 153.90 |
| sacoma-3745 | 3,497.51 | $\leq 0.01$ | 14,261.22 | 13,993.96 | 55.21 | 307.75 |
| jardim-campinas-2485 | 2,676.44 | $\leq 0.01$ | 6,005.80 | 5,880.30 | 14.54 | 124.39 |
| jardim-centenario-1117 | 1,493.66 | $\leq 0.01$ | 3,023.03 | 2,934.86 | 1.28 | 102.39 |
| aeroporto1-1301 | 3,275.88 | $\leq 0.01$ | 6,981.83 | 6,906.52 | 21.26 | 113.12 |
| baixa-grande-2223 | 2,507.89 | $\leq 0.01$ | 5,400.10 | 5,278.40 | 6.67 | 115.32 |
| limoeiro-3472 | 3,710.09 | $\leq 0.01$ | 13,752.99 | 13,460.23 | 84.87 | 270.69 |
| vila-amelia-1112 | 920.98 | $\leq 0.01$ | 1,803.76 | 1,764.32 | 0.57 | 95.85 |
| parque-peruche-710 | 810.28 | $\leq 0.01$ | 1,563.50 | 1,516.50 | 0.27 | 92.95 |
| jardim-japao-879 | 1,822.58 | $\leq 0.01$ | 2,922.80 | 2,887.17 | 0.60 | 60.36 |
| favela-dos-anjos-922 | 997.80 | $\leq 0.01$ | 2,395.28 | 2,303.53 | 0.69 | 140.05 |
| campo-belo-2515 | 3,195.24 | $\leq 0.01$ | 8,766.53 | 8,582.25 | 12.46 | 174.36 |
| jardim-tres-marias-3282 | 3,231.24 | $\leq 0.01$ | 14,540.45 | 14,109.31 | 42.47 | 349.99 |
| itaim-paulista-3209 | 3,299.92 | $\leq 0.01$ | 12,119.09 | 11,868.96 | 59.08 | 267.25 |
| **Avg.** | 2,685.93 | $\leq 0.01$ | 7,829,94 | 7,671,73 | 27.30 | 170.85 |

objective values when used alone. This result is expected because Guard Shift only modifies the locations of non-drivable units. In contrast, the other two movements directly modify patrol routes and therefore have a stronger impact on the solution.

Figure 9 presents the results for the larger instances limoeiro-3472 and vila-renato-645. The same behavior observed in small instances is found in larger problems, although the differences between configurations become clearer. For limoeiro-3472, the combined configuration achieves the best objective value, outperforming all others. Among the pairwise combinations, Path Switch + Guard Shift provides the best performance, followed by Loop Insertion + Guard Shift and Path Switch + Loop Insertion, which produce similar results. A similar pattern is observed for vila-renato-645, where the combined configuration again yields the best solution. The pairwise combinations yield slightly lower but still competitive results, while the individual movements produce weaker solutions. As in the small instances, Loop Insertion and Path Switch show stronger individual performance than Guard Shift, reinforcing the idea that route modification plays a key role in improving the generated patrol routes. Nevertheless, the inclusion of Guard Shift in combined configurations improves overall results by better aligning non-drivable units with uncovered areas.

The sensitivity analysis reveals three important aspects of the PROTECTOR heuristic. First, the best performance is consistently obtained when all three neighborhood movements are applied together. Second, the Path Switch and Loop Insertion movements are responsible for most of the improvements, as they directly modify patrol routes. Finally, Guard Shift plays a complementary role by improving the spatial distribution of non-drivable units after route adjustments. Overall, these results justify adopting the full neighborhood configuration in PROTECTOR.

# 7 Concluding Remarks

The PROTECTOR heuristic, presented in this paper, offers an effective strategy for tackling the Routing of Police Vehicles in Large Urban Centers (RVP-Urb) problem. The main goal of the proposed heuristic is to maximize crime coverage by using the available police resources (*e.g.*, police officers, vehicles, *etc*) in an efficient and well-balanced way. Unlike most existing approaches discussed in Section 3, PROTECTOR is specifically designed to reflect the organizational structure of Brazilian police departments, in which urban areas are subdivided into smaller zones. Preventive patrol routes are therefore generated for each of these zones rather than for the entire city simultaneously. Moreover, PROTECTOR leverages crime hotspot data within a specified time interval to construct routes that maximize coverage of locations with a higher likelihood of criminal activity. Since hotspots evolve, PROTECTOR can be reinvoked periodically to update and adapt

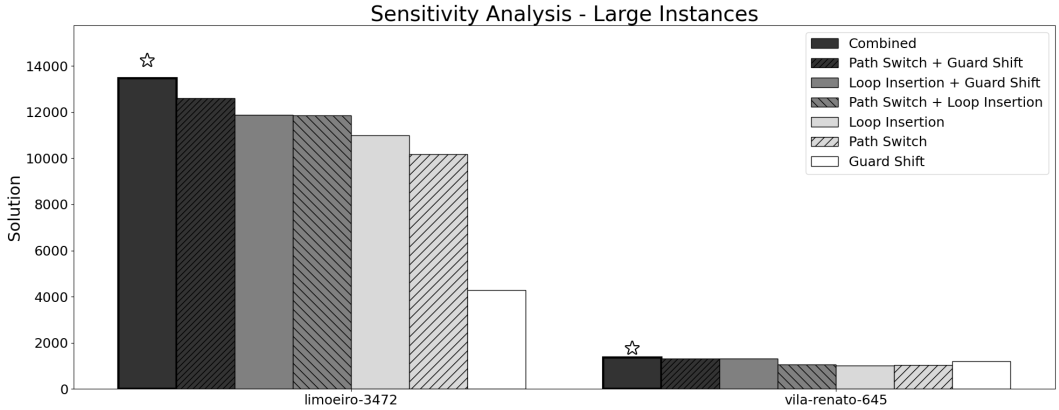


**Figure 9.** sensitivity Analysis of PROTECTOR for large-scale instances.

the preventive patrol routes in response to the most recent crime patterns.

Several experiments were conducted to evaluate the performance of PROTECTOR, all based on real-world data obtained from the PolRoute-DS dataset [Cunha Sá *et al.*, 2022]. This dataset aggregates crime events provided by the Military Police of the State of São Paulo and represents them within a graph structure corresponding to São Paulo's street network. It is worth mentioning that PROTECTOR demonstrated satisfactory results in generating preventive police patrol routes, both when compared with the optimal results obtained from the mathematical model and when compared with routes generated by domain experts. In most instances, PROTECTOR achieved the best possible solution, and in cases where the mathematical model could not produce an exact solution, due to excessive computational time or system limitations such as memory constraints, it produced high-quality approximate solutions.

As future work, we plan to integrate the proposed PROTECTOR heuristic with data mining techniques to identify and analyze patterns among the generated route segments, thereby uncovering potential insights into patrol behavior and spatial crime dynamics. In addition, we plan to implement alternative heuristics grounded in well-established metaheuristic frameworks, such as Simulated Annealing and Tabu Search.

## Funding

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. The authors would like to thank FAPERJ and CNPq.

## Authors' Contributions

All authors contributed to the conception of this study. EC, IR and DO performed the experiments. All authors read and approved the final manuscript.

## Competing interests

The authors declare that they have no competing interests.

## Availability of data and materials

The datasets generated and/or analyzed during the current study are available in our GitHub repository `https://github.com/UFFeScience/protector`.

# References

Alikhademi, K., Drobina, E., Prioleau, D., Richardson, B., Purves, D., and Gilbert, J. E. (2022). A review of predictive policing from the perspective of fairness. *Artif. Intell. Law*, 30:1–17. DOI: 10.1007/s10506-021-09286-4.

Bertsimas, D., van Ryzin, G., and Management, S. (1991). A stochastic and dynamic vehicle routing problem in the euclidean plane. *Operations Research*, 39:601–615. DOI: 10.1287/opre.39.4.601.

Bilal, M., Usmani, R. S. A., Tayyab, M., Mahmoud, A. A., Abdalla, R. M., Marjani, M., Pillai, T. R., and Targio Hashem, I. A. (2020). *Smart Cities Data: Framework, Applications, and Challenges*, pages 1–29. Springer International Publishing, Cham. DOI: 10.1007/978-3-030-15145-$4_6 - 1$.

Boba, R. (2005). Crime analysis and crime mapping. DOI: 10.4324/9780202101148-18.

Borrion, H., Ekblom, P., Alrajeh, D., Borrion, A. L., Keane, A., Koch, D., Mitchener-Nissen, T., and Toubaline, S. (2019). The problem with crime problem-solving: Towards a second generation pop? *The British Journal of Criminology*, 60(1):219–240. DOI: 10.1093/bjc/azz029.

Chawathe, S. S. (2007). Organizing hot-spot police patrol routes. In *2007 IEEE Intelligence and Security Informatics*, pages 79–86. DOI: 10.1109/ISI.2007.379538.

Chen, H., Cheng, T., and Wise, S. (2015). Designing daily patrol routes for policing based on ant colony algorithm. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-4/W2:103–109. DOI: 10.5194/isprsannals-II-4-W2-103-2015.

Chen, H., Cheng, T., and Wise, S. (2017). Developing an online cooperative police patrol routing strategy. *Computers, Environment and Urban Systems*, 62:19–29. DOI: 10.1016/j.compenvurbsys.2016.10.013.

Coelho, V. N., Coelho, I. M., Oliveira, T. A., and Ochi, L. S. (2019). *Smart and Digital Cities - From Computational*

*Intelligence to Applied Social Sciences*. Springer. DOI: 10.1007/978-3-030-12255-3.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. The MIT Press, 2nd edition. DOI: 10.1017/cbo9781107279667.013.

Cunha Sá, B., Muller, G., Banni, M., Santos, W., Lage, M., Rosseti, I., Frota, Y., and de Oliveira, D. (2022). Polroute-ds: a crime dataset for optimization-based police patrol routing. *Journal of Information and Data Management*, 13(1). DOI: 10.5753/jidm.2022.2355.

Dewinter, M., Vandeviver, C., Vander Beken, T., and Witlox, F. (2020). Analysing the police patrol routing problem: A review. *ISPRS International Journal of Geo-Information*, 9(3). DOI: 10.3390/ijgi9030157.

Dos Anjos Júnior, O. R., Lombardi Filho, S. C., and Maia do Amaral, P. V. (2018). Determinantes da criminalidade na região sudeste do brasil: uma aplicação de painel espacial. *Econ. Soc. Territ.*, (57). DOI: 10.22136/est20181176.

Duarte, A., Sánchez-Oro, J., Mladenović, N., and Todosijević, R. (2018). Variable neighborhood descent. In Martí, R., Pardalos, P. M., and Resende, M. G. C., editors, *Handbook of Heuristics*, pages 341–367. Springer. DOI: 10.1007/978-3-319-07153-4$_9 - 1$.

Eysenck, H. J. and Gudjonsson, G. H. (2013). *The causes and cures of criminality*. Springer Science & Business Media. DOI: 10.5860/choice.27-0986.

García-Ponce, O. and Laterzo, I. (2023). The legacy of mexico's drug war on youth political attitudes. 2023(96). DOI: 10.35188/UNU-WIDER/2023/404-5.

Gendreau, M., Laporte, G., and Semet, F. (2006). The maximal expected coverage relocation problem for emergency vehicles. *J. Oper. Res. Soc.*, 57(1):22–28. DOI: 10.1057/palgrave.jors.2601991.

Hale, C. D. and Chapman, S. G. (1981). *Police patrol: Operations and management*. Wiley NJ. Book.

Hjalmarsson, R., Machin, S., and Pinotti, P. (2024). Chapter 9 - crime and the labor market. volume 5 of *Handbook of Labor Economics*, pages 679–759. Elsevier. DOI: 10.1016/bs.heslab.2024.11.008.

Inmon, W., Welch, J., and Glassey, K. (2005). *Building the Data Warehouse.*, volume 39. Sons Inc, New York. DOI: 10.1145/240455.240470.

Kim, D., Kan, Y., Aum, Y., Lee, W., and Yi, G. (2023). Hotspots-based patrol route optimization algorithm for smart policing. *Heliyon*, 9(10). DOI: 10.1016/j.heliyon.2023.e20931.

Lenstra, J. K. and Rinnooy Kan, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11:221–227. DOI: 10.1002/net.3230110211.

Lourenço, V., Mann, P., Guimaraes, A., Paes, A., and de Oliveira, D. (2018). Towards safer (smart) cities: Discovering urban crime patterns using logic-based relational machine learning. In *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, pages 1–8. IEEE. DOI: 10.1109/IJCNN.2018.8489374.

López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Stützle, T., and Birattari, M. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58. DOI: 10.1016/j.orp.2016.09.002.

M. Ball, B. L. G. and Bodin, L. D. (1983). Planning for truck fleet size in the presence of a common□carrier option. *Decision Sciences*, 14(1):103–120. DOI: 10.1111/j.1540-5915.1983.tb00172.x.

Melo, A., Belchior, M., and Furtado, V. (2005). Analyzing police patrol routes by simulating the physical reorganization of agents. In Sichman, J. S. and Antunes, L., editors, *MABS 2005*, volume 3891 of *Lecture Notes in Computer Science*, pages 99–114. Springer. DOI: 10.1007/11734680_8.

Obagbuwa, I. C. and Abidoye, A. P. (2021). South africa crime visualization, trends analysis, and prediction using machine learning linear regression technique. *Appl. Comput. Intell. Soft Comput.*, 2021:5537902:1–5537902:14. DOI: 10.1155/2021/5537902.

Oliveira, L. F., Oliveira, D., and Frota, Y. (2023). Defining routes for emergency response from climate events: a data-oriented approach. *IEEE Latin America Transactions*, 21(10):1064–1072. DOI: 10.1109/TLA.2023.10255445.

Oliveira, W. A. d., Moretti, A. C., and Reis, E. F. (2015). Multi-vehicle covering tour problem: Building routes for urban patrolling. *Pesqui. Oper.*, 35(3):617–644. DOI: 10.1590/0101-7438.2015.035.03.0617.

Reis, D., Melo, A., Coelho, A. L. V., and Furtado, V. (2006). Towards optimal police patrol routes with genetic algorithms. In Mehrotra, S., Zeng, D. D., Chen, H., Thuraisingham, B. M., and Wang, F., editors, *IEEE ISI*, volume 3975 of *Lecture Notes in Computer Science*, pages 485–491. Springer. DOI: 10.1007/11760146_45.

Resende, M. G. C. and Ribeiro, C. C. (2016). *Optimization by GRASP: Greedy Randomized Adaptive Search Procedures*. Springer Publishing Company, Incorporated. DOI: 10.1007/978-1-4939-6530-4.

Ristvej, J., Lacinák, M., and Ondrejka, R. (2020). On smart city and safe city concepts. *Mob. Networks Appl.*, 25(3):836–845. DOI: 10.1007/s11036-020-01524-4.

Saint-Guillain, M., Paquay, C., and Limbourg, S. (2021). Time-dependent stochastic vehicle routing problem with random requests: Application to online police patrol management in brussels. *Eur. J. Oper. Res.*, 292(3):869–885. DOI: 10.1016/j.ejor.2020.11.007.

Samanta, S., Sen, G., and Ghosh, S. K. (2022). A literature review on police patrolling problems. *Annals of Operations Research*, 316(2):1063–1106. DOI: 10.1007/s10479-021-04167-0.

Shen, Y., Lee, J., Jeong, H., Jeong, J., Lee, E., and Du, D. H. C. (2018). SAINT+: Self-adaptive interactive navigation tool+ for emergency service delivery optimization. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1038–1053. DOI: 10.1109/TITS.2017.2710881.

Silva, L. J. S., Fiol-González, S., Almeida, C. F. P., Barbosa, S. D. J., and Lopes, H. (2017). Crimevis: An interactive visualization system for analyzing crime data in the state of rio de janeiro. In Hammoudi, S., Smialek, M., Camp, O., and Filipe, J., editors, *ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems, Volume 1, Porto, Portugal, April 26-29, 2017*, pages 193–200. SciTePress. DOI: 10.5220/0006258701930200.

Sá, B., Muller, G., Banni, M., Santos, W., Lage, M., Rosseti, I., Frota, Y., and de Oliveira, D. (2021). Polroute-ds: um dataset de dados criminais para geração de rotas de patrulhamento policial. In *Anais do III Dataset Showcase Workshop*, pages 117–127, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/dsw.2021.17420.

Tregle, B., Smith, M. R., Fahmy, C., and Tillyer, R. (2025). More than meets the eye: examining the impact of hot spots policing on the reduction of city-wide crime. *Crime Science*, 14(1):3. DOI: 10.1186/s40163-025-00247-9.

Turchan, B. and Braga, A. A. (2024). The effects of hot spots policing on violence: A systematic review and meta-analysis. *Aggression and Violent Behavior*, 79:102011. DOI: 10.1016/j.avb.2024.102011.