Optimal and Heuristic Solutions for Efficient Vehicle Patrol Scheduling in Dynamic Urban Safety

Majid Ghasemi
Dept. of Computer Science
Wilfrid Laurier University
Waterloo, Ontario, Canada
mghasemi@wlu.ca

Ebrahim Sarkhouh
Dept. of Computer Science
Wilfrid Laurier University
Waterlo, Ontario, Canada
esarkhouh@wlu.ca

Fadi Alzhouri
Dept. of Elec. & Computer Eng.
GUST University
Mishref, Kuwait
alzhouri.f@gust.edu.kw

Dariush Ebrahimi
Dept. of Computer Science
Wilfrid Laurier University
Waterloo, Ontario, Canada
debrahimi@wlu.ca

2025 IEEE 34th International Symposium on Industrial Electronics (ISIE) | 979-8-3503-7479-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISIE62713.2025.11124793

***Abstract*—Efficient vehicle patrol scheduling is crucial for enhancing urban safety, ensuring security, and optimizing operational efficiency. Traditional scheduling methods often struggle to balance multiple objectives under constraints such as limited vehicles and patrol times. Addressing these challenges can improve response times, reduce operational costs, and maintain effective security coverage. In this work, we propose a twofold solution: an optimization model designed for small-scale problems to guarantee optimal solutions, and two heuristic approaches—Adaptive Hill Climbing-Based Patrol Scheduling (AHBPS) and Genetic-Based Dynamic Vehicle Patrol Scheduling (GDVPS)—to address more complex, dynamic scenarios. Our results demonstrate significant improvements in both efficiency and coverage, with GDVPS outperforming AHBPS in more intricate networks. These findings highlight the potential for GDVPS as a dynamic, robust, and scalable solution, warranting further development to enhance urban safety systems.**

I. Introduction

Patrolling is essential for monitoring business operations and security services, aiming to deter criminal activities while safeguarding people and property. Professional services employ highly trained personnel and advanced technologies, such as GPS tracking and real-time reporting, tailored to diverse environments. Vehicle Patrol Scheduling (VPS), a variant of the Vehicle Routing Problem (VRP) introduced in [1], focuses on scheduling patrols for public safety, emphasizing efficient area coverage and route unpredictability. While the VRP seeks to optimize vehicle routes for deliveries, considering constraints like capacity and time windows, it has evolved into various variants. Solving VPS presents challenges due to its NP-hardness and constraints such as shift durations, rest periods, and variable travel times. Patrol vehicles must maximize visits and revisit locations at specific intervals, necessitating scalable optimization techniques for real-time adjustments in large urban networks [2,3].

The literature reveals several existing approaches to VPS, which can be categorized into five main areas. First, studies on optimizing patrol scheduling in urban transit systems focus on maximizing coverage and minimizing predictability using mathematical models [4,5]. Second, police patrol scheduling is addressed through mathematical programming [6,7]. Third, maritime patrol scheduling, which optimizes boat schedules using column generation techniques, is explored in [8]–[10]. Fourth, incident response routing has been studied, with heuristics proposed for timely responses [11,12]. Lastly, improvements to bus scheduling using a multi-objective genetic algorithm are presented in [13].

In this paper, we address the VPS problem, where a fleet of vehicles is tasked with patrolling a set of locations within a designated area during a specified operational period (e.g., 12 hours from 8:00 PM to 8:00 AM). The goal is to maximize the total number of locations patrolled while adhering to several constraints: First, each location requires a minimum patrol time, meaning a vehicle must spend a certain amount of time at the location (e.g., 5 minutes). Second, there is a maximum allowable gap between consecutive visits to any location (e.g., 30 minutes). Third, each vehicle must return to the depot for rest after operating for a defined period (e.g, 2 hours). Lastly, each vehicle must meet a minimum rest time requirement before resuming operations (e.g., 10 minutes). To address this problem, we first formulate the problem mathematically as an optimization model to provide optimal solutions for small-scale, static scenarios. To effectively address larger, dynamic environments, we develop a scalable hill-climbing heuristic method called Adaptive Hill Climbing-Based Patrol Scheduling (AHBPS), which could be executed regularly over time or after any changes happen in the environment. Additionally, we introduce a Genetic-based method, Genetic-Based Dynamic Vehicle Patrol Scheduling (GDVPS), which enhances the optimization process, yielding near-optimal solutions for complex dynamic urban networks.

We evaluate the performance of AHBPS and GDVPS through extensive simulations on real-world urban maps, considering various constraints such as network size, patrol time, rest periods, and different travel times between locations. A comprehensive comparison of solutions generated by the optimization model, AHBPS, and GDVPS demonstrates the scalability and effectiveness of our proposed methods. Our key finding indicates that GDVPS consistently outperforms AHBPS in terms of scalability and solution quality, particularly in complex scenarios involving large urban networks. While the optimization model provides optimal solutions for smaller instances, its scalability limitations render GDVPS the preferred approach for real-world applications in urban safety operations. By integrating heuristic and genetic algorithm approaches, we offer a practical and effective solution for maximizing patrol coverage while adhering to real-world constraints.

The remainder of this paper is organized as follows. Section II presents the system model and problem description, followed by the mathematical formulation in Section III. Section IV details the AHBPS method, while Section V explains the GDVPS approach. In Section VI, we provide performance evaluations. Section VII concludes the paper and summarizes our findings.

II. System Model and Problem Definition

We conceptualize the operational environment for a fleet of patrol vehicles as a weighted directed graph $G = (N, E)$, where N represents the set of nodes, which includes a depot $l \in N = 0$, and a set of locations $l = \{1, 2, ..., |N|\}$ that must be patrolled regularly. Each location has a minimum patrolling time T^{patrol} and a maximum allowable gap between visits T^{gap}. The set E signifies the links representing possible routes between these nodes, with weights corresponding to travel time between locations, denoted as $W_{l,l'}$, which are measured based on current traffic conditions. These weights are dynamically updated to reflect the real-world nature of the problem. During the operational period, vehicles in the fleet ($c \in C$) are systematically assigned routes within the graph. Each route begins at the depot, visits designated locations, and returns to the depot upon completing its shift. Each vehicle must adhere to a mandatory rest time T^{rest} after a maximum operational duration of T^{shift}. This structure ensures comprehensive coverage of all areas of interest during the operational time. The objective of this study is to optimize the routing of vehicles in a network to maximize the total visits to all patrolled locations for securing within the designated operational period. This involves planning routes from the depot to various locations and back while ensuring that mandatory rest is taken at the depot. Our primary goal is to increase the total number of locations visited throughout the workday. To achieve this, we ensure no more than one vehicle visits the same location simultaneously and that each location is revisited within a specified gap to maintain continuous safety.

Problem Definition (maximizing the number of visited locations): *Given a graph $G = (N, E)$, where N and E respectively represent the set of nodes and links, let there be a set of locations $l \in N = 1, 2, ..., |N|$. The problem is to determine the optimal route for each vehicle $c \in C$ during each shift, starting from the depot l=0, while visiting the maximum number of locations and returning to the depot. This must be done while considering the maximum operation time for each shift T^{shift}, rest time T^{rest}, minimum location patrolling time t^{patrol}, and the maximum allowable gap between visits T^{gap}. The objective is to maximize the total number of locations visited during the entire operational period.*

III. Mathematical Formulation

In this section, we mathematically formulate the problem to obtain the optimal solutions. The used notations are listed in Table I. Let V_l be the total number of visits to location l. The objective function of the problem is to maximize this number for all locations and can be mathematically written as:

TABLE I
Notations used for the problem formulation

Decision Variables	
$V_l \geq 0$	Total number of visits to location l.
$X_{c,l,v} \in \{0, 1\}$	Indicates whether vehicle c has done its v-th visit at location l.
$Y_{c,l,v,s} \in \{0, 1\}$	Indicates whether vehicle c has done its v-th visit at location l at s-th shift.
$A_{c,v} \geq 0$	Time when vehicle c starts its v-th visit.
$P_{c,v} \geq 0$	Patrolling time of vehicle c on its v-th visit.
$L_{l,s} \geq 0$	Time when location l has been visited on s-th shift.
Parameters	
T^{shift}	Maximum operation time at each shift.
$W_{l,l'}$	Travel Time between locations l and l'.
$Adj_{l,l'}$	Indicating the adjacency of locations l and l'.
T^{rest}	Minimum rest time of a vehicle.
D	The depot location index.
T^{patrol}	Minimum patrolling time of a location.
T^{gap}	Maximum gap between consecutive visits.
R_c^{start}	Start of the rest time of vehicle c.
R_c^{end}	End of the rest time of vehicle c.

$$\text{Maximize} \quad \sum_{l=1}^{|N|} V_l \tag{1}$$

Subject to: (2)-(16), where these constraints are introduced in detail below.

Initial Location Constraint: Each vehicle c should start its journey from the depot (D) as the initial starting point. If we let $X_{c,l,v}$ to indicate whether vehicle c has done its v-th visit at location l, then the initial starting location for each vehicle can be written as:

$$X_{c,D,1} = 1 \qquad \forall c \in C \tag{2}$$

Operation shift Constraint: If we let $Y_{c,l,v,s}$ indicate that vehicle c has done its v-th visit at location l at s-th shift, then the following constraint ensures that if vehicle c has not done its v-th visit at location l, then this visit did not take place during any shift. That is, if $X_{c,l,v}$ is zero, so is $Y_{c,l,v,s}$.

$$Y_{c,l,v,s} \leq X_{c,l,v} \qquad \forall c \in C, l \in N, v = 1..\boldsymbol{v}^{\max}, s = 1..\boldsymbol{s}^{\max} \tag{3}$$

Single Vehicle Constraint: As enforced by the following constraint, a vehicle can only visit one location at a time.

$$\sum_{l \in N} X_{c,l,v} = 1 \qquad \forall c \in C, v = 1..\boldsymbol{v}^{\max} \tag{4}$$

Single Location Constraint: No location can be patrolled by multiple vehicles at the same time as enforced by the following:

$$\sum_{c \in C} \sum_{v=1}^{\boldsymbol{v}^{\max}} Y_{c,l,v,s} \leq 1 \qquad \forall l \in N, s = 1..\boldsymbol{s}^{\max} \tag{5}$$

Path Feasibility Constraint: A vehicle cannot travel between two locations unless a path exists. This constraint ensures that the vehicle's movements are feasible within the road network.

$$X_{c,l,v} \leq \sum_{l' \in N} X_{c,l',v-1} \cdot Adj_{l,l'} \forall c \in C, v = 1..\boldsymbol{v}^{\max}, l \& l' \in N \tag{6}$$

Depot Visit Constraint: A vehicle should visit the depot at least once during each shift. This ensures that vehicles return to the depot for maintenance, refueling, rest, etc.

$$\sum_{v=1}^{\boldsymbol{v}^{\max}} \sum_{s=1}^{\boldsymbol{s}^{\max}} Y_{c,\mathrm{D},v,s} \geq 1 \quad \forall c \in C \tag{7}$$

Depot Rest Time Constraint: A vehicle must stay at the depot for a duration of T^{rest} to ensure both vehicles and personnel receive adequate rest before resuming patrol. If we let $P_{c,v}$ to be the patrolling time of vehicle c on its v-th visit, then it must be at least T^{rest} when the vehicle is at the depot.

$$P_{c,v} \geq T^{rest} \cdot \sum_{s=1}^{\boldsymbol{s}^{\max}} Y_{c,D,v,s} \quad \forall c \in C, v = 1..\boldsymbol{v}^{\max} \tag{8}$$

Vehicle Visit Time Constraint*: In the following, we calculate the time when vehicle c starts its v-th visit ($A_{c,v}$), which can be obtained by summing the time when the vehicle started its previous visits ($A_{c,v-1}$), patrolling time ($P_{c,v-1}$), and the time needed to travel from location l to l', denoted as $W_{l,l'}$.

$$A_{c,v} = A_{c,v-1} + P_{c,v-1} + \sum_{l \in N} \sum_{l' \in N} W_{l,l'} \cdot X_{c,l,v-1} \cdot X_{c,l',v} \quad \forall c \in C, v = 1..\boldsymbol{v}^{\max} \tag{9}$$

Rest Time End Constraint*: The time when vehicle c finishes its rest, denoted as R_c^{end}, must occur after the vehicle enters the depot and completes its required rest period.

$$\sum_{s=1}^{\boldsymbol{s}^{\max}} y_{c,D,v,s} \cdot (A_{c,v} + P_{c,v}) \leq R_c^{end} \quad \forall c \in C, v = 1..\boldsymbol{v}^{\max} \tag{10}$$

Rest Time Start Constraint: The rest time for a vehicle, denoted by R_c^{start}, begins when it arrives at the depot.

$$A_{c,v} \geq R_c^{start} \cdot \sum_{s=1}^{\boldsymbol{s}^{\max}} Y_{c,D,v,s} \quad \forall c \in C, v = 1..\boldsymbol{v}^{\max} \tag{11}$$

Maximum Shifts Constraint: A location cannot be visited and patrolled more than the number of possible shifts.

$$\sum_{c \in C} \sum_{v=1}^{\boldsymbol{v}^{\max}} Y_{c,l,v,s} = (s < V_l) \qquad \forall l \in N, s = 1..\boldsymbol{s}^{\max} \tag{12}$$

where $(s < V_l)$ is a boolean expression that evaluates to 1 if the condition is true, and 0 otherwise.

Vehicle Arrival Constraint: Constraint (13) ensures that the patrolling of location l by vehicle c during the s-th shift cannot start until the vehicle has arrived at the location. Here, $L_{l,s}$ indicates when location l was visited during the s-th shift.

$$Y_{c,l,v,s} \cdot A_{c,v} \leq L_{l,s} \quad \forall c \in C, l \in N, v = 1..\boldsymbol{v}^{\max}, s = 1..\boldsymbol{s}^{\max} \tag{13}$$

Consecutive Visits Gap Constraint*: This constraint ensures that a location cannot remain unpatrolled for longer than a duration of T^{gap}, where T^{gap} represents the maximum allowable gap between consecutive visits. In particular, the arrival time of vehicle c to location l during its v-th visit in the s-th shift, when $Y_{c,l,v,s} = 1$, must occur after the start time of the $(s-1)$-th shift (i.e., $L_{l,(s-1)}$), plus the patrolling time T^{patrol}, and gap time T^{gap}).

$$A_{c,v} \geq Y_{c,l,v,s} \cdot (L_{l,(s-1)} + T^{gap} + T^{patrol}) \quad \forall c \in C, l \in N, v = 1..\boldsymbol{v}^{\max}, s = 2..\boldsymbol{s}^{\max} \tag{14}$$

Minimum Patrolling Duration Constraint*: Each location visited must be patrolled for a minimum duration of T^{patrol} to ensure sufficient security. In the following constraint, the arrival time of vehicle c plus the patrolling duration must occur after the start of patrolling and the minimum allowable patrol time T^{patrol}.

$$A_{c,v} + P_{c,v} \geq (T^{patrol} + L_{l,s}) \cdot Y_{c,l,v,s} \quad \forall c \in C, l \in N, v = 1..\boldsymbol{v}^{\max}, s = 1..\boldsymbol{s}^{\max} \tag{15}$$

Shifts Order Constraint: The patrolling shifts must occur in order, and the difference between them must exceed the minimum patrol time (T^{patrol}) plus the gap time (T^{gap}).

$$L_{l,s} \geq L_{l,s-1} + (s < V_l) \cdot (T^{patrol} + T^{gap}) \quad \forall l \in N, s = 1..\boldsymbol{s}^{\max} \tag{16}$$

Constraints marked with "*" have not been linearized due to space limitations. However, it is important to emphasize that the OP model, including its linearized form, was fully implemented and tested.

IV. The Heuristic Method: AHBPS

In this section, we propose a heuristic algorithm designed to efficiently handle both small and large instances of the NP-hard VPS problem. The AHBPS (Adaptive Hill-climbing Based Patrol Scheduling) method, detailed in Alg. 1, consists of two main components: the *Routing Process* and the *Patrol Scheduling Process*. The *Routing Process*, outlined in Alg. 2, seeks to determine the most efficient patrol routes and vehicle shifts, while the *Patrol Scheduling Process*, described in Alg. 3, ensures vehicles follow an optimized schedule for maximum coverage efficiency.

The AHBPS method (Alg. 1) takes as input the graph $G(N, E)$, the vehicle fleet C, depot location D, number of shifts S, maximum operation time per shift T^{shift}, minimum patrol time T^{patrol}, and travel times $W_{l,l'}$. The outputs of this algorithm are the visited locations and routes (paths) for vehicles.

To determine routes, AHBPS calls the `RouteFinding` function in line 3, detailed in Alg. 2, which outputs the patrol locations, $Visited_{locations}$, based on the optimal solution found. The function is initiated by simulating travel time fluctuations, introducing random variations to reflect real-world conditions (lines 2-5). These fluctuations adjust the travel times $W_{l,l'}$ between locations l and l' (line 5). The primary solution method uses Hill Climbing (lines 6-10),

Algorithm 1: The Heuristic Method: AHBPS

1 **Input:** $G(N, E)$, C, D, S, T^{shift}, T^{patrol}, $W_{l,l'}$
Output: $Visited_{locations}$, $routes$
2 Initialize graph G, vehicles C, depots D, shifts S, shift time T^{shift}, patrol time T^{patrol}, and travel times $W_{l,l'}$;
3 $Visited_{locations} \leftarrow RouteFinding(G, C, D, S, T^{shift}, T^{patrol}, W_{l,l'})$;
4 $L_{l,s}, routes \leftarrow ShiftManagement(G, C, D, S, T^{shift}, T^{patrol}, W_{l,l'})$;
5 **Return** $Visited_{locations}$, $routes[s][c]$;

Algorithm 2: Routing Process

1 **Function** `RouteFinding`$(G, C, D, S, T^{shift}, T^{patrol}, W_{l,l'})$**:**
2   **if** *a random value < 0.25* **then**
3   **foreach** $(l, l') \in G$ **do**
4       $fluctuation \leftarrow$ random value of ± 2;
5       $W_{l,l'} \leftarrow W_{l,l'} + fluctuation$;
6   **while** *Improvement found* **do**
7     $neighbors \leftarrow$ Generate neighboring solutions from CS;
8     $CSO_{better} \leftarrow$ Evaluate and select the best neighbor using updated $W_{l,l'}$;
    if $(CSO_{better} > CSO)$ **then**
9       $CS \leftarrow$ Update Current Solution;
10       $CSO \leftarrow CSO_{better}$;
11   **foreach** $location \in CS$ **do**
12   **if** *location is visited in the best solution* **then**
13       $Visited_{locations} \leftarrow Visited_{locations} \cup \{location\}$;
14   **return** $Visited_{locations}$;

where neighboring solutions are generated from the current solution (line 8), and the best solution is selected if it improves the objective (lines 9-10). This iterative process continues until no further improvements are achievable. Finally, for each location in the current solutions (CS), if it is optimal, the function adds it to the visited locations set. Alg. 2 thus seeks optimal patrol routes by balancing shift constraints, travel times, and real-time fluctuations, enhancing patrol efficiency.

AHBPS then calls the `ShiftManagement` function (line 4) to ensure each location is visited according to the vehicle routes. Alg. 3 handles the patrol scheduling process, aiming to enable vehicles to follow an optimal schedule across multiple shifts. The process begins by initializing key parameters, such as the current time (CT), the last visit time for each location ($L_{l,s}$), and patrol routes. The algorithm checks if a location requires revisiting using the `Revisit` function (lines 2-6), ensuring no location remains unpatrolled within a given interval. The `ShiftManagement` function (lines 7-18) manages vehicle route selection during each shift. For each

Algorithm 3: Patrol Scheduling Process

1 **Parameters Initialization:**
$CT, L_{l,s}, LL, SST, SET, routes, CS, CSO$;
2 **Function** `Revisit`$(CT, L_{l,s}, LL)$**:**
3   **if** $(CT - L_{l,s}) \leq 30$ *AND* $(!LL$ *OR* $LL \leq CT)$ **then**
4   **return** *True*;
5   **else**
6   **return** *False*;
7 **Function** *ShiftManagement*$(S, C, L_{l,s}, LL, SST, SET, routes, CS, CSO, W_{l,l'})$**:**
8   **for** $s \in S$ **do**
9   **for** $c \in C$ **do**
10   **while** $CT + T^{patrol} \leq SET$ **do**
11         $c_{route} \leftarrow$ Select the next best location;
12   **if** `Revisit`$(CT, L_{l,s}, LL)$ **then**
13           $travel_{time} \leftarrow W_{l,l'}[current_{location}, c_{route}]$;
14           $CT \leftarrow CT + travel_{time} + T^{patrol}$;
15           $routes[s][c] \leftarrow$ Update the route with the visited location;
16           $L_{l,s} \leftarrow$ Update Last Visit Time for the location;
17   $routes[C] \leftarrow D$;
18   **return** $L_{l,s}$, $routes[s][c]$;

vehicle in each shift, if a location needs to be revisited, it selects the next optimal location for a vehicle to visit based on adjacency and travel time, ensuring all locations are patrolled within the allocated shift time T^{shift}. The route is updated with each new visit (line 11). If there is a need to revisit a location, travel time $travel_{time}$, current time CT, and routes for each shift and vehicles ($routes[s][c]$) get updated. The last visit time $L_{l,s}$ (line 16) is recorded for each location. Alg. 3 ensures each location is patrolled within the required time frame, minimizing gaps between visits while optimizing vehicle routes for efficient coverage during patrol shifts. The final output of AHBPS (line 5 in Alg. 1) includes the set of visited locations and the routes taken by the vehicles in every shift, providing a complete patrol plan.

V. Genetic Vehicle Patrol Scheduling (GDVPS)

To further enhance the performance of the AHBPS algorithm, we introduce a robust meta-heuristic, Genetic-based Dynamic Vehicle Patrol Scheduling (GDVPS). Fig. 1 provides an overview of the GDVPS framework applied in our study, which demonstrates the rationale behind crossover, selection, fitness computation, and mutation.

The GDVPS algorithm builds on the initial patrol schedules generated by AHBPS, refining them through an evolutionary approach to identify the most efficient patrol schedules. The algorithm begins by creating an initial population of patrol

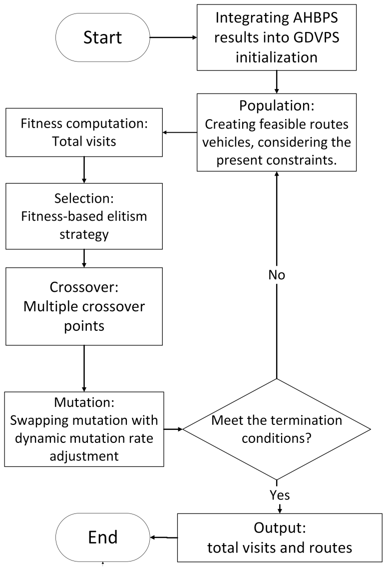


Fig. 1. GDVPS Flowchart

schedules. Some schedules in this population derive from the AHBPS outputs, providing a strong baseline, while the remainder are randomly generated to introduce diversity and enable exploration of alternative routing options. Each schedule defines the routes and timings for all vehicles across their designated shifts. Each schedule's fitness is evaluated based on its effectiveness in maximizing the coverage of unique locations. This objective aligns with the goal of optimizing patrol coverage across as many locations as possible.

The top-performing schedules are then selected for crossover, where components of different schedules are combined to generate new, improved patrol routes. To further diversify solutions and prevent convergence on suboptimal routes, the algorithm also applies mutations, small random alterations, such as swapping the order of visited locations for each vehicle. These genetic operations (selection, crossover, and mutation) are iteratively applied across multiple generations until there is no improvement or a maximum generation count is reached. The final result is the optimal patrol schedule identified, maximizing the number of unique locations visited while ensuring all constraints are met.

Unlike general heuristic methods such as Ant Colony Optimization, GDVPS uses tailored genetic operators specifically designed to accommodate the unique constraints of the VPS problem, such as multi-shift schedules, vehicle allocations, rest periods, and patrol duration. This customization enables GDVPS to generate feasible, constraint-compliant solutions that are difficult to achieve with conventional heuristic methods.

VI. Performance Evaluation

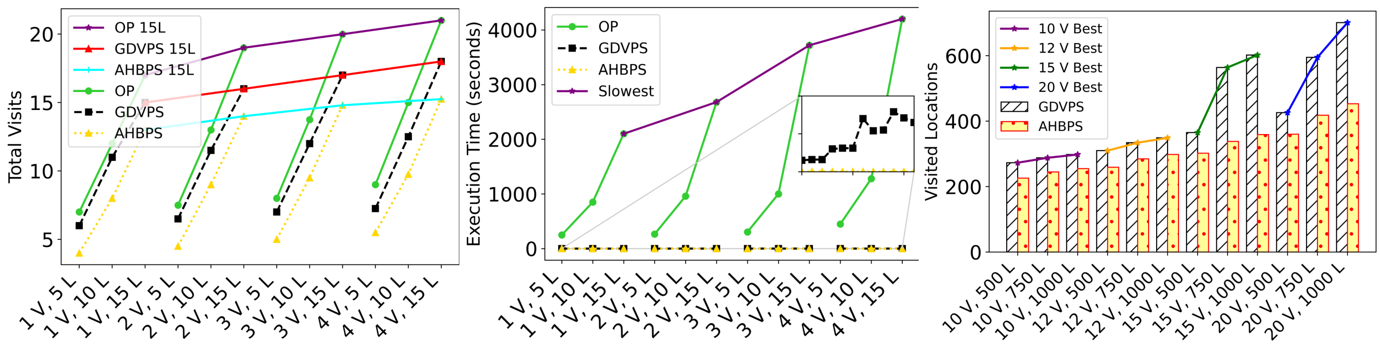
In this section, we evaluate the proposed methods, Optimum (OP), AHBPS, and GDVPS, on both small and large complex networks. The primary performance metric is the total visits. To generate the graphs, we utilize real-world maps extracted from OpenStreetMap (OSM). By assigning varying numbers of nodes to represent locations, we create diverse graphs featuring different target locations (e.g., banks) within the same map size. This approach allows us to generate numerous real-world scenarios, serving as distinct instances to evaluate proposed methods. We employ the OSMNX library in Python to construct the graphs from the extracted maps, which we then use as inputs for our methods. Travel times between nodes are calculated based on the distances represented in the generated maps. To simulate real-time traffic conditions, we assign weights to the links according to a normal distribution with a mean of 15 minutes and a standard deviation of 5 minutes. To incorporate real-time dynamics, such as varying traffic densities (e.g., incidents that may extend travel time), we vary travel times by ± 2 minutes in 25% of the runs. This accounts for scenarios like low traffic (reduced travel time) or unforeseen incidents (increased travel time).

This methodology effectively mitigates randomness, providing a clearer, more accurate reflection of algorithm performance across varying conditions and underscoring the efficiency of the methods. Initially, we fix certain parameters while testing various instances, and after analyzing the results, we adjust inputs to assess their impact. Unless specified otherwise, the rest time is set to 10 minutes, patrol time per location is 5 minutes, and the travel time distribution maintains a mean of 15 minutes. Additionally, results are averaged over 100 runs.

We evaluate the performance of our proposed heuristics, AHBPS and GDVPS, against the optimal solution (OP) across varying numbers of vehicles and locations. As shown in Fig. 2(a), GDVPS consistently outperforms AHBPS, particularly in more complex networks with 15 locations, demonstrating its robustness and effectiveness. However, both AHBPS and GDVPS exhibit diminished effectiveness and fail to achieve the optimal solution as expected, since the OP approach excels by thoroughly exploring all potential configurations, ensuring that the absolute best result is identified without overlooking any possibilities, thereby guaranteeing the highest level of efficiency, and prune suboptimal regions.

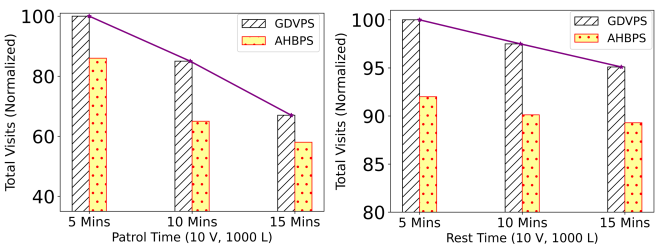
Fig. 2(b) illustrates the execution times for the proposed methods. The OP approach requires significantly longer execution times due to its exhaustive search, with execution time growing exponentially as network size increases. AHBPS consistently executes in approximately 0.01 seconds across all instances, while GDVPS takes around 0.3 seconds on average. The OP approach becomes impractical for instances with more than 15 locations due to excessive computation time, attributed to the NP-hard nature of the problem. These findings indicate that while the OP is theoretically superior, it is not feasible for larger problems. GDVPS emerges as a more scalable and robust alternative to AHBPS in complex scenarios. While AHBPS remains competitive in simpler cases, it struggles as the complexity increases.

We further evaluated AHBPS and GDVPS on larger networks with varying numbers of vehicles and locations, as illustrated in Fig. 2(c). GDVPS consistently outperforms AHBPS in nearly all instances, particularly as the number of locations increases, demonstrating better scalability with



(a) (b) (c)

Fig. 2. Comparing the performance (total visits) & execution time of different methods (OP(Optimum), AHBPS, and GDVPS) – (a) total visits for small network , (b) execution time comparison, (c) total visits for large networks



(a) (b)

Fig. 3. Performance comparison of AHBPS vs. GDVPS (total visits) for 10 vehicles across 1,000 locations, varying (a) patrol time and (b) rest period.

problem complexity. Both methods show improvement in visiting more locations as the problem size grows; however, the performance gap widens in favor of GDVPS. This advantage arises from the genetic algorithm's ability to efficiently explore and exploit the search space through mechanisms such as crossover and mutation, which enhance the process of finding optimal or near-optimal solutions. As a result, GDVPS adapts more effectively to increasing complexity, maintaining higher performance levels compared to AHBPS as the number of locations expands. We also analyzed how different parameters affect the objective of maximizing visited locations. Using GDVPS on large networks with 10 vehicles and 1,000 locations, we examine the effects of changes in rest periods and patrol times. First, we compare different patrol durations, 10 and 15 minutes, against the baseline value of 5 minutes. As expected, longer patrol times result in fewer visited locations, as vehicles spend more time at each location (see Fig. 3(a)). Next, we assessed the impact of varying rest periods at the depot by adjusting the initial 10 minutes to 5 and 15 minutes (Fig. 3(b)). Reducing the rest time to 5 minutes yields a slight improvement, while increasing it to 15 minutes diminishes performance. However, shorter rest periods may have unexamined psychological effects, so adequate rest is recommended. In conclusion, GDVPS proves to be highly effective in both small and large networks and demonstrates efficiency in real-time applications due to its near-optimal performance.

VII. Conclusion

We addressed the critical Vehicle Patrol Scheduling problem in urban safety operations, aiming to balance area coverage with patrol efficiency. Using a dual approach, we first formulated the problem mathematically to derive optimal solutions for small, static instances and then developed two heuristic methods, Adaptive Hill Climbing-Based Patrol Scheduling (AHBPS) and Genetic-Based Dynamic Vehicle Patrol Scheduling (GDVPS), for larger, dynamic scenarios. GDVPS utilizes crossover and mutation operators to efficiently explore vast solution spaces while maintaining diversity and scalability. Extensive simulations demonstrated that GDVPS significantly outperforms AHBPS in complex networks, achieving superior coverage and operational efficiency. Thus, GDVPS is a robust and scalable solution for large-scale urban safety operations.

References

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management science*, vol. 6, no. 1, pp. 80–91, 1959.

[2] B. B. Keskin, S. R. Li, D. Steil, and S. Spiller, "Analysis of an integrated maximum covering and patrol routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 1, pp. 215–232, 2012.

[3] N. Adler, A. S. Hakkert, J. Kornbluth, T. Raviv, and M. Sher, "Location-allocation models for traffic police patrol vehicles on an interurban network," *Annals of operations research*, vol. 221, pp. 9–31, 2014.

[4] H. C. Lau and A. Gunawan, "The patrol scheduling problem." PATAT, 2012.

[5] H. C. Lau, Z. Yuan, and A. Gunawan, "Patrol scheduling in urban rail network," *Annals of Operations Research*, vol. 239, pp. 317–342, 2016.

[6] S. Yan, C.-Y. Wang, and Y.-W. Chuang, "Optimal scheduling for police patrol duties," *Journal of the Chinese Institute of Engineers*, vol. 43, no. 1, pp. 1–12, 2020.

[7] N. Adler, A. S. Hakkert, T. Raviv, and M. Sher, "The traffic police location and schedule assignment problem," *Journal of Multi-Criteria Decision Analysis*, vol. 21, no. 5-6, pp. 315–333, 2014.

[8] P. Chircop, T. Surendonk, M. van den Briel, and T. Walsh, "A column generation approach for the scheduling of patrol boats to provide complete patrol coverage," in *Proceedings of the 20th international congress on modelling and simulation*, 2013, pp. 1–6.

[9] P. A. Chircop, T. J. Surendonk, M. H. van den Briel, and T. Walsh, "On routing and scheduling a fleet of resource-constrained vessels to provide ongoing continuous patrol coverage," *Annals of Operations Research*, vol. 312, no. 2, pp. 723–760, 2022.

[10] P. A. Chircop and T. J. Surendonk, "On integer linear programming formulations of a patrol boat scheduling problem with complete coverage requirements," *The Journal of Defense Modeling and Simulation*, vol. 18, no. 4, pp. 429–439, 2021.

[11] L. Hajibabai and D. Saha, "Patrol route planning for incident response vehicles under dispatching station scenarios," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 1, pp. 58–70, 2019.

[12] S. Choi, M. Lee, H. Park, and J. Han, "Mathematical programming-based heuristic for highway patrol drone scheduling problem," *Socio-Economic Planning Sciences*, vol. 93, p. 101907, 2024.

[13] X. Zuo, C. Chen, W. Tan, and M. Zhou, "Vehicle scheduling of an urban bus line via an improved multiobjective genetic algorithm," *IEEE Transactions on intelligent transportation systems*, vol. 16, no. 2, pp. 1030–1041, 2014.