ABSTRACT

Title of Dissertation: MATHEMATICAL PROGRAMMING
AND HEURISTIC APPROACHES FOR
THREE NOVEL VEHICLE ROUTING PROBLEMS

Yuchen Luo
Doctor of Philosophy, 2022

Dissertation Directed by: Professor Bruce Golden
The Robert H. Smith School of Business

Vehicle routing problems (VRPs) arise in key real-world domains including mail delivery, police patrol, and snow plowing. Due to the inherent complexity of the VRP, no algorithm currently exists that can solve the VRP exactly in polynomial time. In the meantime, high quality solutions are needed to tackle challenging issues such as rising mail and package delivery costs and surging crime rates. In this dissertation, we design efficient routing solutions for three applications of the VRP.

First, we consider the application of the hot spot police patrol, where hot spots refer to locations with high crime rates. We examine the routing problem for a patrol car that visits locations in a region and remains close to the center of the region modeling a high-crime neighborhood (HCN). We formulate this problem as the traveling salesman problem with a center (TSPC), in which we minimize an energy function including L (the length of a tour) and C (the distance from a tour to the center, defined by some metric). To address the TSPC, we propose a metric to

measure C rather accurately and also introduce the idea of a triangular path, in which the patrol car no longer travels in a straight line between two nodes. We show that under identical circumstances, the tour with triangular paths remains closer to the center than a tour using all direct edges in both a Euclidean graph and a grid network.

Second, we extend the hot spot police patrol to a fleet of patrol cars. Given the disorder and chaos in the HCN, we require at least one patrol car in the HCN at any given time during the patrol. We call this routing problem the hot spot coverage patrol problem (HSCPP). In the HSCPP, the importance of a patrol location is quantified by a prize and our objective is to maximize the sum of prizes collected by the patrol cars while obeying all operations requirements. We propose mathematical formulations and develop several solution approaches for the HSCPP. These solution approaches are based on integer programming and column generation. We then conduct a detailed computational study to compare these approaches using real crime data from Montgomery County, Maryland. We point out the best solution approaches in terms of optimality gap, runtime, and workload balancing.

Third, we consider mail deliveries. To reduce the fleet size and carbon emissions of a postal service, we allow two mail carriers per truck. We call this problem the paired mail carrier problem (PMCP). Our objective is to minimize the route completion time while ensuring that each service stop is fully serviced and obeys all feasibility constraints. We develop a mixed integer programming formulation and two fast heuristics for the PMCP. We evaluate the impact of the paired mail carriers (PMC) setting on both a one-truck situation and a fleet (multiple trucks) situation, relative to the single mail carrier (SMC) setting. Overall, the PMC setting can not only accomplish over 50% more work within the same shift hours but can also lead to a 22% cost savings.

MATHEMATICAL PROGRAMMING AND HEURISTIC APPROACHES FOR THREE NOVEL VEHICLE ROUTING PROBLEMS

by

Yuchen Luo

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2022

Advisory Committee:
Professor Bruce Golden, Chair/Advisor
Professor Ilya Ryzhov
Professor Paul Schonfeld
Professor Paul Smith
Professor Rui Zhang

© Copyright by
Yuchen Luo
2022

Dedicated to my family.

Acknowledgments

First and foremost, I would like to thank my advisor Professor Bruce Golden for his outstanding guidance and tremendous support on this thesis. Throughout my PhD study, he has given me detailed and timely advice on research and job searching and in many other areas. His patience and motivation helped me get through the tough times in the past three years. I feel privileged to have had the opportunity to work with him. I hope one day I can be a great mentor to others as Professor Golden has been to me.

I would also like to acknowledge two other mentors, as this thesis would not be possible without their help. I am grateful to Professor Rui Zhang, who has worked closely with me on all three projects of this thesis. Whenever I have had questions about research or life in general, he has sacrificed his time to meet with me immediately. I would also like to thank Professor Stefan Poikonen, who has given me many actionable research ideas. I deeply appreciate his insights and suggestions on various topics of this thesis.

I would like to express my gratitude to Professor Ilya Ryzhov, Professor Paul Schonfeld, and Professor Paul Smith for donating their invaluable time to serve on my committee and give me feedback on this thesis.

The PhD program in the Applied Mathematics and Scientific Computing (AMSC) program has been a wonderful place for me. I especially want to thank the AMSC coordinator, Jessica Sadler, for giving prompt replies to all my questions and helping me with the administrative

details. I would also like to thank my friends at Maryland for their help and support throughout the journey.

Last but not least, I would like to give my sincere thanks to my mother, Hong Yuan, and my father, Huide Luo. None of this would be possible and meaningful without their support and encouragement. They have given me the best education opportunities and taught me to become a better person.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

API	Application Programming Interface
BPS	Best Partition Solution
CG	Column Generation
DOIs	Dual-optimal Inequalities
DP	Dynamic Programming
ESPPRC	Elementary Shortest Path Problem with Resource Constraints
HCN	High-crime Neighbourhood
HSCPP	Hot Spot Coverage Patrol Problem
IP	Integer Programming
LHS	Left-hand Side
LKH	Lin-Kernighan-Helsgaun
LP	Linear Programming
MCPRP	Maximum Coverage and Patrol Routing Problem
MIP	Mixed Integer Programming
PMC	Paired Mail Carrier
PMCP	Paired Mail Carrier Problem
RHS	Right-hand Side
SMC	Single Mail Carrier
TOP	Team Orienteering Problem
TSP	Traveling Salesman Problem
TSPC	Traveling Salesman Problem with a Center
USPS	United States Postal Service
VRP	Vehicle Routing Problem

Chapter 1: Introduction

Logistics businesses often need to deliver massive amounts of goods to customers each day. It is common for goods to experience multiple modes of transport during shipping [2], i.e., they are transported to a cargo hub by air, rail, or sea and then delivered to the customers by truck. In particular, last-mile deliveries conducted by trucks are important in terms of speedy deliveries and customer experience [53]. Businesses make huge investments building routing software, which gives trucks turn-by-turn navigation. For example, UPS spends nearly one billion a year on its routing optimization software [6]. On the other hand, the trucking routing problem is becoming harder to solve for two reasons. First, retailers require their orders to be delivered accurately and quickly [36]. This leads to tighter constraints in the routing problem. Second, the shipping volume increases exponentially as e-commerce emerges. It is estimated that Amazon delivers 1.6 million packages a day [9]. This increases the scale of the routing problem massively.

Routing problems also arise in other domains including garbage collection, police patrol, and snow plowing. For instance, waste management companies assign trucks to follow a set of routes to collect garbage, such that all bins are emptied and the waste is driven to disposal sites [10]. Police departments design patrol routing for their vehicle fleets to prevent offenders from committing crimes and to respond to real-time incidents in a timely way [16]. Municipalities plan snowplow truck routes to remove snow and ice on roads. Efficient routing could reduce

costs significantly and improve public safety [32]. Although these routing problems may have different objectives (minimize costs, response time, or duration), they can all be categorized as combinatorial optimization problems, which often suffer from the curse of dimensionality.

The traveling salesman problem (TSP) is one of the most well-studied optimization problems. One has to find the shortest tour through a number of locations in a region. The TSP is NP-hard, i.e., the computational complexity grows exponentially as the number of locations increases. The vehicle routing problem (VRP) is a generalization of the TSP and finds the optimal routes in terms of length for a fleet of vehicles. In addition, the VRP often incorporates complicating factors such as time windows, backhauls, and travel distance constraints to solve the routing for a particular application. For example, when considering the police patrol routing problem, a location often has a time window when accidents are prone to happen [31]. As a result, the length of the tour is not the sole concern and covering the location within the time window is equally important. Due to the NP-hardness of the TSP and VRP, the size of the problem that can be solved optimally is already limited. With these extra complicating factors, the routing problem becomes even more intractable.

In this dissertation, we design routing solutions for three applications of the VRP. First, we consider the application of the hot spot police patrol, where hot spots are known as locations with high crime rates. Lately, researchers suggest that focusing on hot spots during the patrol could alleviate crime problems [7, 8]. Given a region, we have a vehicle patrolling a set of patrol locations. Within the region, there also exists a high-crime neighborhood (HCN) including hot spots. We intend to find the patrol tour which is short in length and close to the HCN. To this purpose, we focus on building an accurate metric for the distance between a tour and the HCN and minimize the distance to the HCN with the metric.

For the second application, we extend the hot spot police patrol to a fleet of vehicles. To patrol the HCN effectively, the cooperation between police vehicles is important [16]. We pay attention to designing a set of routes such that at least one vehicle is always close to the HCN at any given time during the patrol. Therefore, when a crime happens, officers can respond to it promptly. In addition, the presence of police in the HCN can reduce crime problems and increase the confidence of the local population. [3, 34].

For the third application, we consider mail deliveries. In recent years, climate change has become an important social issue [4]. Many companies are committed to reducing the carbon footprint and reaching carbon neutrality [47]. For logistics companies, one of the biggest sources of carbon emissions comes from delivery trucks [26]. To reduce the number of trucks, we allow two mail carriers per truck and compare the paired mail carriers (PMC) setting to the traditional single mail carrier (SMC) setting in terms of the completion time of the deliveries and costs to the company. In all three applications, we use integer programming to find the exact solution. When integer programming is no longer effective for larger instances, heuristics are applied to find high-quality solutions in a timely manner.

In Chapter 2, we examine the routing problem for a patrol vehicle that visits locations in a region and remains close to the center of the region modeling an HCN. We formulate this problem as the traveling salesman problem with a center (TSPC) [41], in which we minimize an energy function including L (the length of a tour) and C (the distance from a tour to the center, defined by some metric). To address the TSPC, we propose a metric to measure C rather accurately and also introduce the idea of a triangular path, in which the vehicle no longer travels in a straight line between two nodes. We incorporate a constraint on the travel distance of the patrol vehicle and solve the constrained version of the TSPC with integer programming (IP) for instances of up

to 80 patrol locations. We show that under identical circumstances, the tour with triangular paths remains closer to the center than a tour using all direct edges both in a Euclidean graph and in a grid network.

In Chapter 3, we study the routing problem for a fleet of patrol cars patrolling a region with an HCN. Considering the disorder and chaos in the HCN, we require at least one patrol car in the HCN at any given time during the patrol. We call this routing problem the Hot Spot Coverage Patrol Problem (HSCPP). In the HSCPP, the importance of a patrol location is quantified by a prize. Our objective is to maximize the sum of prizes collected by the patrol cars while obeying all operation requirements. We propose mathematical formulations and develop several solution approaches for the HSCPP. The global approach consists of finding the routing for all patrol cars with a single integer programming (IP) formulation. The partition approach involves first partitioning the region geographically and solving the routing problem in each subregion with two IP formulations. Next, we strengthen the partition approach by developing a column generation (CG) approach, where the initial columns of the CG approach are the solutions generated from the partition approach. In the last solution approach, we use the best partition solution as a warm start for the global approach. We compare these solution approaches in terms of optimality gap, runtime, and workload balancing based on real crime data from Montgomery County, Maryland.

In Chapter 4, we introduce the paired mail carrier problem (PMCP), which allows for two mail carriers per truck. In the PMCP, we have a truck making several stops along a mail route. The objective is to minimize the time from the start to the end of the route while ensuring that each service stop is fully serviced and obeys all feasibility constraints. Given two mail carriers per truck, the work allocation between mail carriers is nontrivial. When one mail carrier is servicing a stop, the other mail carrier has the option of waiting or moving forward to service future stops.

We offer several propositions to limit the search space, which greatly improves the runtime of our solution approaches. We develop a mixed integer programming (MIP) formulation and two fast heuristics for the PMCP. The MIP formulation obtains optimal solutions within minutes for routes with fewer than 40 stops. Further, our heuristics can handle larger instances (100 stops) within one second and obtain high quality solutions. More importantly, we evaluate the impact of the PMC setting on both a one-truck situation and a fleet (multiple trucks) situation, relative to the single mail carrier SMC setting. Overall, the PMC setting can not only accomplish over 50% extra work within the same shift hours but can also lead to a 22% cost savings. We also discuss an extension where we can have three or more mail carriers per truck.

In Chapter 5, we provide concluding remarks and direction for future work.

Chapter 2: A Fresh Look at the Traveling Salesman Problem with a Center

2.1 Introduction

The traveling salesman problem is one of the most well-studied optimization problems. It involves finding the shortest closed tour joining a set of locations. Due to its inherent complexity, no exact algorithm currently exists that can solve the TSP exactly in polynomial time. Instead, robust heuristics like the Lin-Kernighan-Helsgaun (LKH) heuristic are designed to handle situations in which large instances need to be solved. The TSP has numerous applications such as package delivery, drilling holes in circuit boards, genome sequencing, making travel itineraries, our particular topic of interest, police patrol [23], and many others. A police vehicle is often routed within a city to patrol a set of locations, so that a police officer can arrive as quickly as possible in the event of a call for help [39]. Naturally, it is preferred that the police vehicle can remain close to high-crime (centrally located) neighborhoods. Moreover, the patrol route needs to be changed regularly to deter premeditated crimes [16]. Thus, it is also important to be able to compute a variety of effective routes. In addition to patrol cars on the ground, there may be police drones in the air that complement the work of patrol cars on the ground.

In this chapter, we examine the routing problem for a patrol car that visits n locations in a region and remains close to a center of the region modelling the HCN. The locations that need to be visited are nodes and the paths connecting them are the edges. The majority of the

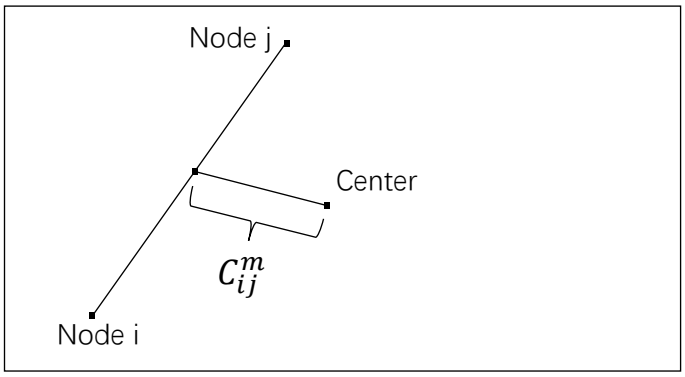


Figure 2.1: Midpoint measure: C_{ij}^m is the distance from the midpoint of edge (i, j) to the center.

chapter assumes that the network is a Euclidean graph with the Euclidean distance as the measure of distance. This assumption is particularly valid for patrols conducted in the air or on water. Further, the center is assumed to be the geometric center of the region for the Euclidean graph. Hence, there are n nodes plus a center in total. In addition to that, we generalize our work to a simple, undirected, connected grid network with the shortest path distance as the measure of distance. Rather than fixing the center as the geometric center, the center could be any node in the grid network, i.e., one of the nodes will function both as the visiting location and the center. The grid network setup is more suitable for ground-based police patrol. We formulate this problem as the traveling salesman problem with a center as defined in Lipowski and Lipowska [41]. The solution to the TSPC is defined as the tour that minimizes the energy function $E = L + rC$, where L is the length of a tour and C is an aggregate measure of distance from an edge in a tour to a center. In particular, $C = \sum_{(i,j)\in T} C_{ij}$, where T is a tour and C_{ij} is a measure of distance from edge (i, j) to the center. r is a parameter that controls the trade-off between L and C.

Lipowski and Lipowska [41] consider the center as the geometric center and propose the midpoint measure for C (see Figure 2.1), where C_{ij}^m is the distance from the midpoint of edge

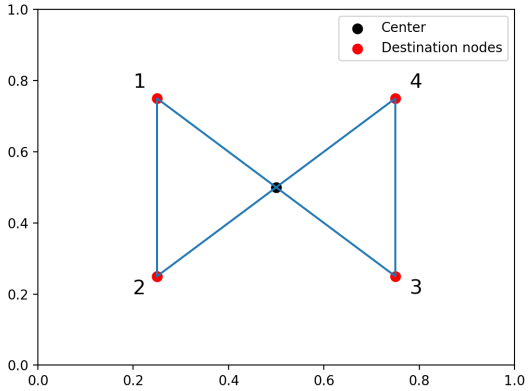


Figure 2.2: A TSPC tour with four nodes. The coordinates of the four nodes are $(0.25, 0.25)$, $(0.25, 0.75)$, $(0.75, 0.25)$, and $(0.75, 0.75)$. Edges $(1, 3)$ and $(2, 4)$ have their midpoints located at the center.

(i, j) to the center. Since the region of interest is the unit square, the center has coordinates $(1/2, 1/2)$. Specifically,

$$C_{ij}^m = \frac{1}{2}\sqrt{(x_i + x_j - 1)^2 + (y_i + y_j - 1)^2}, \tag{2.1}$$

where (x_i, y_i) are the coordinates of the ith node. They solve the TSPC problem using simulated annealing to find a tour that approximately minimizes the energy function E. When plotting the values of L versus r for these solutions, a phase transition is observed near $r = 2$ with a dramatic increase in L (see Figure 2.7a). This indicates the focus of the minimization in E shifting from L to C. At first glance, this jump seems unexpected. We think, however, that the jump in L is largely attributable to the use of the midpoint measure, C_{ij}^m. In particular, when the midpoint of edge (i, j) is near the center, C_{ij}^m may be small. However, if the edge is long, most points on the edge may be significantly farther from the center than the midpoint of the edge would indicate. This induces a behavior in which edges that pass near the center are often selected, even if they

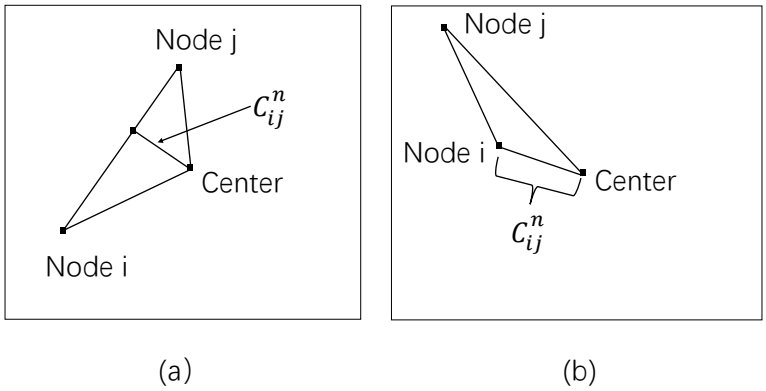
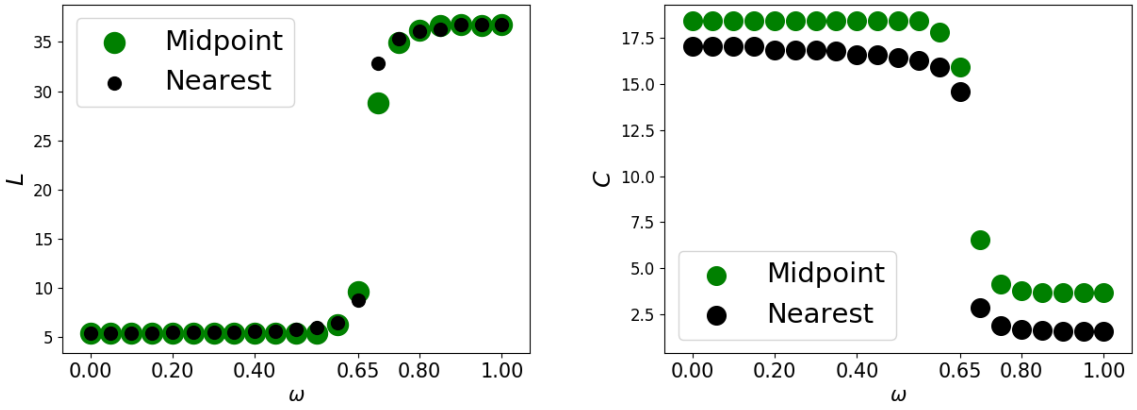


Figure 2.3: Nearest point measure: C_{ij}^n is the shortest distance from edge (i, j) to the center.

are quite long. For instance, as we see in Figure 2.2, $C_{24}^m = 0$ but almost all points on edge $(2, 4)$ have larger distances to the center. In fact, the distances from points on edge $(2, 4)$ to the center range from 0 to $\sqrt{2}/4$ with the average and the median both occurring at $\sqrt{2}/8$. Thus, it might make sense to introduce a different measure of distance to the center that is not focused strictly on the midpoint of the edge.

Alternatively, the nearest point measure is proposed in Price [48], where C_{ij}^n is expressed as the shortest distance from edge (i, j) to the center. In order to visualize C_{ij}^n, we let edge (i, j) be the base of the triangle in Figure 2.3. Then, C_{ij}^n is the height of the triangle when both base angles are acute (see Figure 2.3(a)), whereas C_{ij}^n is the minimum distance between node i to the center and node j to the center if one of the base angles is obtuse (see Figure 2.3(b)). Observe that C_{ij}^n is a lower bound on C_{ij}^m. Thus, when using the nearest point measure, we face the same difficulty as we do when using the midpoint measure to compute C_{ij}. Kistner [33] modifies the energy function to $E^* = (1-\omega)L + \omega C$ for $0 \leq \omega \leq 1$ and solves the TSPC with the midpoint measure and the nearest point measure for different values of ω. Figure 2.4 shows the relationship between L, C and ω for both metrics. It is observed that when $\omega \geq 0.65$, L and C change dramatically for



(a) L vs. ω. (b) C vs. ω.

Figure 2.4: Relationship between L, C and ω for the midpoint measure and the nearest point measure.

both metrics. These findings convince us that relying on only one point per edge when computing C_{ij} will not yield a smooth increase in L as a function of r in E or ω in E^*.

One of the main contributions of this chapter is that we study the TSPC using an improved measure of C_{ij}. Primarily, we introduce an average distance measure, which computes C_{ij} as the average distance over many points on edge (i, j) to the center. Ideally, a true (continuous) average distance measure that computes the distance to the center for all points on the edge might be applied, but it is often too computationally expensive. Thus, we implement a discrete version in which C_{ij} is the average distance to the center for some equidistant points on edge (i, j). For the sake of simplicity, we denote the discrete average distance measure as the average distance measure. Compared to the midpoint measure and the nearest point measure, the average distance measure yields a more realistic C_{ij} by measuring C_{ij} over more points on edge (i, j). Moreover, we introduce two variants of the energy function and aim to smooth the phase transition. Note that the two ends of the phase transition in Figure 2.7a are relatively flat in L due to the one-sided focus on either L or C when minimizing E. On the other hand, the jump in the phase transition is

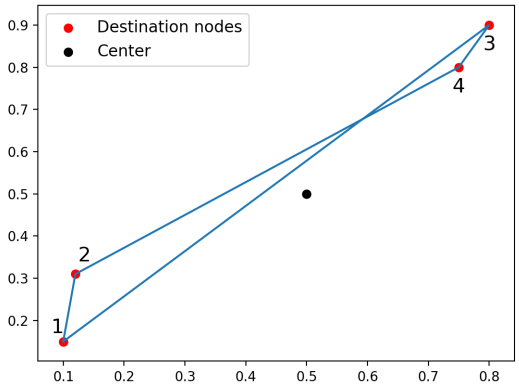


Figure 2.5: A visual illustration of crisscrossing edges and zig-zag edges. Observe that edge $(1, 2)$ and edge $(3, 4)$ are zig-zag edges while edge $(1, 3)$ and edge $(2, 4)$ are crisscrossing edges.

often more interesting since L and C play similar roles in E in determining the optimal tour. In appearance, tours represented by the jump have a fair combination of crisscrossing edges (edges that connect the nodes on opposite sides of the center) and zig-zag edges (edges that link the nodes nearby). Figure 2.5 provides a visual illustration of these two types of edges. For the police patrol application, a smooth transition results in a diverse pool of routes to satisfy the daily needs of police patrol. For instance, the requirement for closeness to the center may vary based on the crime rate of the area. In this case, routes in the pool can be chosen accordingly.

Next, we present our notion of using triangular paths to get closer to the center: The path travels towards the center from a node, reaches a point near to the center, then turns and continues to the next node in the tour. We will later show that under the same circumstances, a triangular path tour is closer to the center than a tour that only contains direct edges, according to our measure of closeness. Further, we redefine the average distance measure and the triangular path in grid networks and show that the triangular path tour still outperforms the direct path tour in the grid network setup.

The remainder of this chapter is structured as follows. In Section 2.2, we define the average distance measure and present the phase transition for the three metrics. We also introduce two variants of the energy function to smooth the phase transition and the triangular path notion. In Section 2.3, we present the constrained TSPC for minimizing C or E subject to a constraint on L. In Section 2.4, we analyze runtime. In Section 2.5, we extend the TSPC to grid networks. Finally, some concluding remarks are offered in Section 2.6.

2.2 A New Approach to the TSPC Problem

In Section 2.2.1, we define the average distance measure for C. We then solve the TSPC problem and identify the phase transitions for the midpoint measure, the nearest point measure, and the average distance measure in Section 2.2.2. In Section 2.2.3, we propose two variants of the energy function. In Section 2.2.4, we introduce the idea of triangular paths between nodes i and j.

2.2.1 Definition of the Average Distance Measure

To obtain a more accurate C_{ij}, we introduce the average distance measure, where we sample $(p-1)$ equidistant points on edge (i,j) (breaking the edge into p segments) and take the average distance to the center over these points as our C_{ij}^a (see Figure 2.6). We denote the coordinates of the kth sampled point on edge (i,j) by (x_{ij}^k, y_{ij}^k), with $1 \leq k \leq p-1$. Then, we have

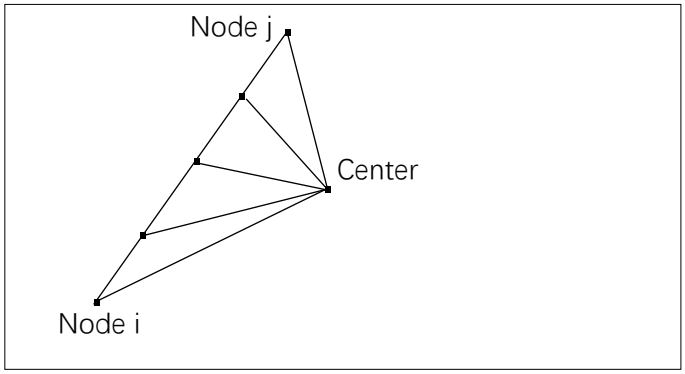
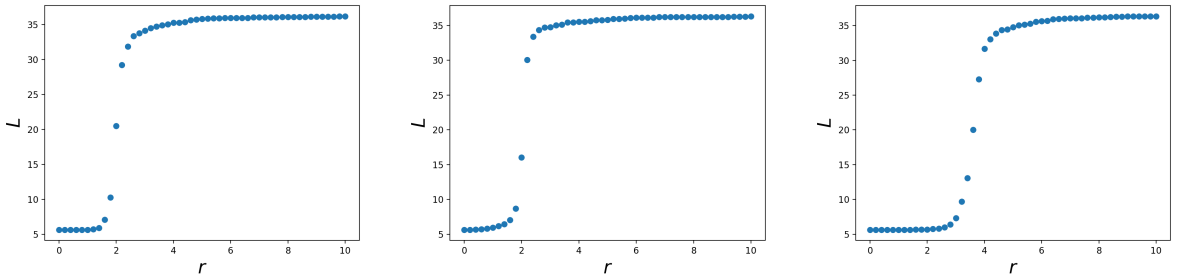


Figure 2.6: Average distance measure. C_{ij}^a is the average distance from sampled points on edge (i, j) to the center. Note that $p = 4$ here.

$$
\begin{aligned}
C_{ij}^a &= \frac{1}{p-1} \sum_{k=1}^{p-1} \left(\sqrt{\left(x_{ij}^k - \frac{1}{2} \right)^2 + \left(y_{ij}^k - \frac{1}{2} \right)^2} \right), \\
\text{where } x_{ij}^k &= \frac{k}{p} x_i + \left(1 - \frac{k}{p} \right) x_j, \\
y_{ij}^k &= \frac{k}{p} y_i + \left(1 - \frac{k}{p} \right) y_j.
\end{aligned}
\tag{2.2}
$$

Note that we only sample the interior points of edge (i, j) here since the two endpoints are usually outliers when computing their distances to the center. As p grows, our C_{ij}^a will converge to the true average $\int_0^1 \left(\sqrt{(ax_i + (1-a)x_j - 1/2)^2 + (ay_i + (1-a)y_j - 1/2)^2} \right) da$. If we apply the average distance measure to the instance in Figure 2.2, we have that $C_{24}^a \approx 0.1767$ when $p = 51$. Compared to the true average of $\sqrt{2}/8 \approx 0.1768$, the relative error for the average distance measure is nearly zero, whereas the relative error for the midpoint measure and the nearest point measure is 100%. On the other hand, for edge $(1, 2)$, we find that $C_{12}^m = 0.25$, $C_{12}^n = 0.25$, and $C_{12}^a \approx 0.287$. Compared to the true average of 0.286948, the relative errors for the metrics are 12.88%, 12.88%, and 0.008%, respectively. Thus, it is fair to say that the average distance



(a) Midpoint measure. (b) Nearest point measure. (c) Average distance measure.

Figure 2.7: Length of the LKH tour L as a function of r for the three metrics.

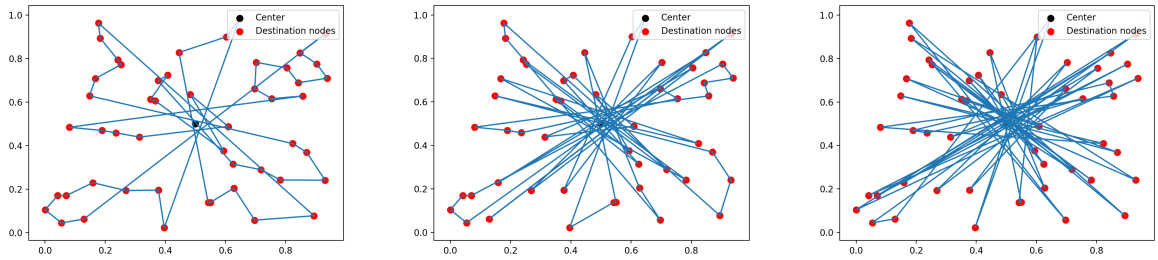
measure gives us a more accurate C compared to the midpoint measure or the nearest point measure. In the meantime, since C_{ij}^a is independent of r, we only need to compute it once. Thus, the computational complexity of the average distance measure is only $O(n^2 p)$.

2.2.2 Phase Transition for the Metrics

2.2.2.1 Midpoint Measure

To begin, we seek to reproduce the phase transition results obtained in Lipowski and Lipowska [41] using the LKH heuristic [29]. Note that LKH is an effective heuristic for solving the TSP. In general, it is faster than simulated annealing and, for instances with $n \leq 100$, it often generates the optimal solution. In particular, LKH 2.0.9 was used to find the near-optimal solution. For the dataset, we generate n random nodes using the standard uniform distribution. Note that our center is $(1/2, 1/2)$ since our region of interest is the unit square. The value of r ranges from 0 to 10 in increments of 0.2. By simulating 10 random instances for $n = 50$, we plot the average length L of the LKH tour as a function of r in Figure 2.7a.

We find that a phase transition occurs around $r = 2$. One can see that the increase in L is initially slow, then gets extreme around $r = 2$, and flattens again after $r > 4$. This abrupt



(a) $r = 1.8$. (b) $r = 2$. (c) $r = 2.2$.

Figure 2.8: LKH tours with the midpoint measure for various values of r.

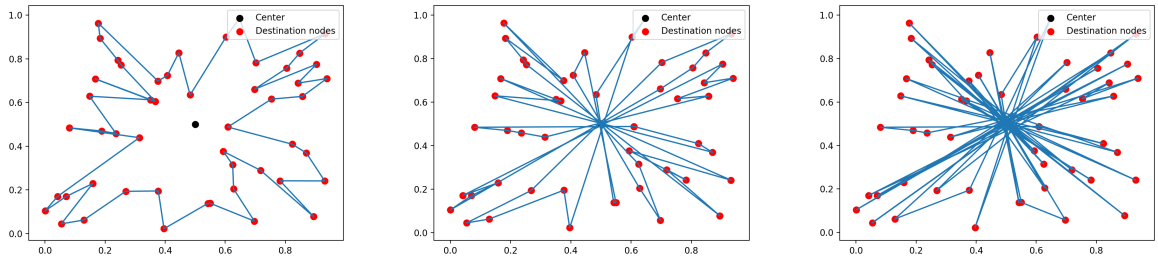
increase is more obvious if we plot the LKH tour (see Figure 2.8). At $r = 1.8$, we are starting to pick up edges across the region in order to keep the midpoints close to the center. Upon increasing r, more crisscrossing edges are included in the tour in order to minimize C. However, this strongly biases the selection of edges in the LKH tour to only those with a midpoint near the center, because, for large values of r, this consideration is dominant. From now on, we denote the phase before the transition as the L-dominated phase and the phase after the transition as the C-dominated phase.

2.2.2.2 Nearest Point Measure

We implement the same experiment using the nearest point measure and plot the phase transition in Figure 2.7b. Compared to the midpoint measure, the jump in L still exists and seems to be at least as severe. The TSPC tours in Figure 2.9 also show extensive crisscrossing as $r \geq 2$.

2.2.2.3 Average Distance Measure

Before implementing the same experiment for the average distance measure, we would like to guarantee the accuracy of C_{ij}^{a} by picking a sufficiently large p. Hence, we plot the average of



(a) $r = 1.8$. (b) $r = 2$. (c) $r = 2.2$.

Figure 2.9: LKH tours with the nearest point measure for various values of r.

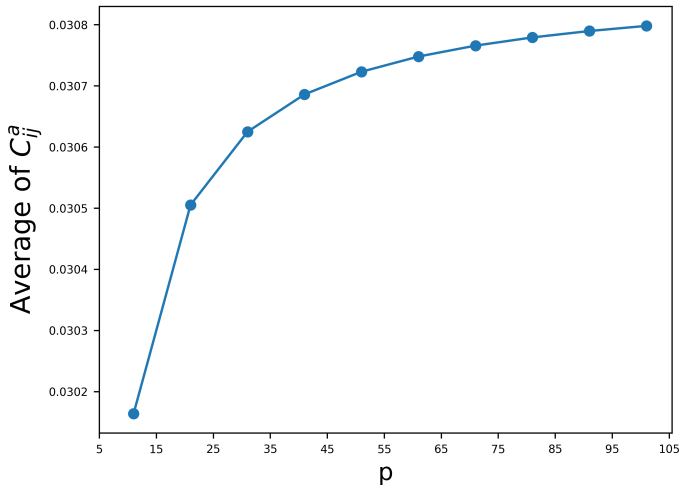
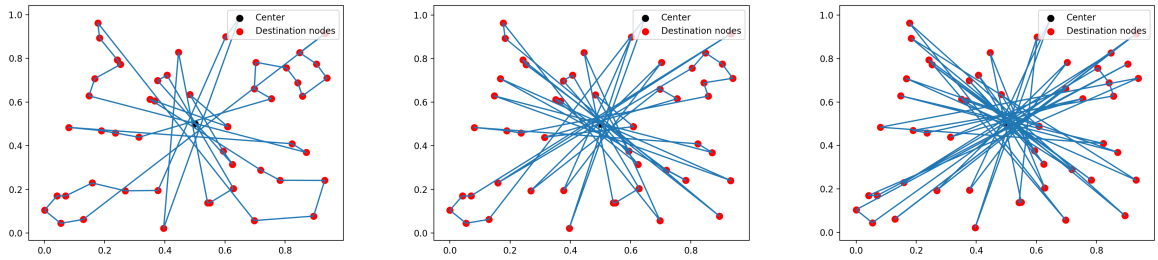


Figure 2.10: Average distance to the center for all possible edges as a function of p.

C_{ij}^a for $1 \leq i, j \leq n = 50$ versus p over 10 random instances in Figure 2.10. We find that the average of C_{ij}^a displays asymptotic behavior, converging to the true (continuous) average distance as p increases. We chose $p = 81$ for our phase transition experiment since the difference in the average of C_{ij}^a between $p = 81$ and $p = 101$ is 1.89×10^{-5}, which is negligible for us. Applying the LKH to the average distance measure, we see that the jump in L has been smoothed in Figure 2.7c, i.e., there are more tours with length L between 10 and 30. In Figure 2.11, we observe similar crisscrossing behavior as we saw with the midpoint measure and the nearest point measure. At this point, we are more convinced that more than one point per edge is required for



(a) $r = 3.4$. (b) $r = 3.6$. (c) $r = 3.8$.

Figure 2.11: LKH tours with the average distance measure for various values of r.

an accurate representation of C and a smooth phase transition. Unless specified, we will use the average distance measure from now on with $p = 81$. Finally, we would like to conclude with the advantages and disadvantages of the three metrics, as shown in Table 2.1.

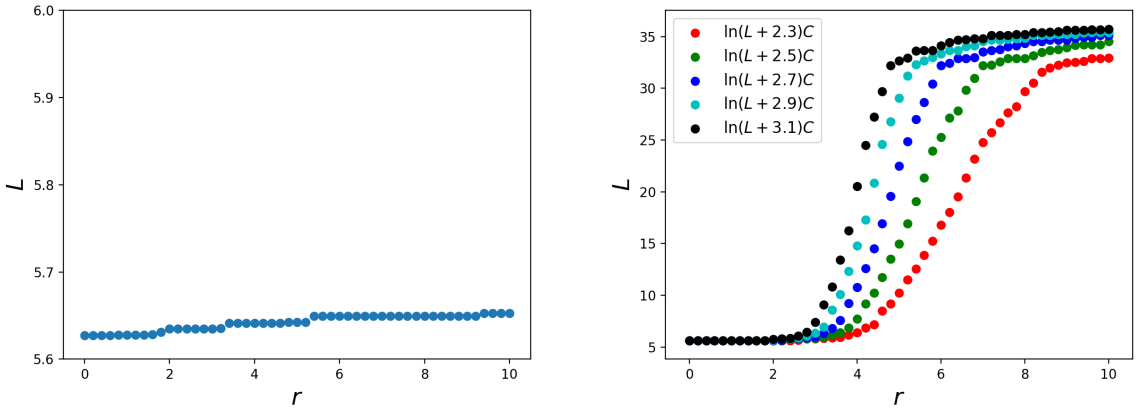
Measures	*Advantages*	*Disadvantages*
Midpoint, Nearest point	Easy to implement	Excessive focus on one point per edge for C
Average Distance	More accurate representation of C	L stays flat longer

Table 2.1: Advantages and disadvantages of the three metrics.

2.2.3 Variants of E

2.2.3.1 $E' = L + rL \cdot C$

Although we have improved the smoothness of the transition by using the average distance measure, the number of tours with length L between 10 and 30 is still limited. This phenomenon results from the excessive focus on C as r increases near and after the transition. Thus, we modify the energy function E to be $E' = L + rL \cdot C$, where $L \cdot C := \sum_{(i,j)\in T} L_{ij}C_{ij}$. By introducing a multiplicative term $L \cdot C$, we hope that E' not only continues to minimize C when r is large, but also alleviates the dramatic increase in L around the transition. In Figure 2.12a, we plot the



(a) Energy function $E' = L + rL \cdot C$.

(b) Energy function $E'' = L + r\ln(L + X) \cdot C$.

Figure 2.12: Phase transition for the energy functions E' and E''.

phase transition for E' over 10 random instances and find that there is no phase transition, i.e., the L-dominated phase tour is always preferred for $0 \leq r \leq 10$.

In order to further understand why E' is stuck in the L-dominated phase, we consider two tours, e.g., tour A and tour B. We denote the length and the distance to the center of tour A by L_A and C_A. The corresponding distances for tour B are L_B and C_B, respectively. Assume that $L_A \leq L_B$ and $C_A \geq C_B$. Thus, tour A could be the TSPC tour when $r = 0$ whereas tour B could be the TSPC tour when $r = 10$. In order to make the transition from tour A to tour B possible, we need $E'(A) \geq E'(B)$, i.e., $(L_A \cdot C_A - L_B \cdot C_B) \geq (L_B - L_A)/r$. Note that $(L_B - L_A)/r$ is non-negative for $L_A \leq L_B$. Hence, for the phase transition to happen, one necessary condition is a non-increasing $L \cdot C$ as we transform from the L-dominated phase to the C-dominated phase. In other words, the minimum of $L \cdot C$ should be obtained in a C-dominated phase tour if the phase transition exists. However, when we minimize $L \cdot C$ for each instance in our experiment using LKH, the resulting optimal tour always corresponds to a TSPC tour before the jump, i.e., an L-dominated phase tour. As a result, one cannot observe the phase transition for E' in Figure

2.12a.

2.2.3.2 $E'' = L + r\ln(L+X)\cdot C$

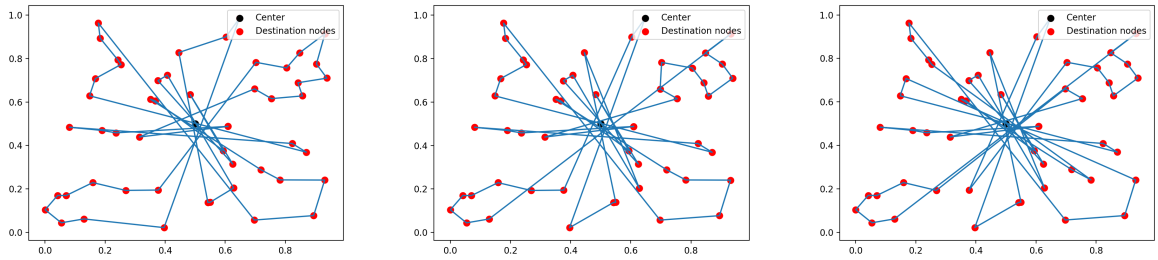
The failure of E' results from the fact that the term $L\cdot C$ reinforces the tendency to minimize L. In order to decrease the dependence on L in E', we change the energy function to $E'' = L + r\ln(L+X)\cdot C$, where $\ln(L+X)\cdot C := \sum_{(i,j)\in T}\ln(L_{ij}+X)C_{ij}$. X is a constant and $X > 1$. This makes sure that $\ln(L_{ij}+X) > 0$ for all $1 \le i,\ j \le n$. We want to see if we can control the phase transition by altering X. For tour A and tour B, the energy functions in the form of E'' are

$$\begin{aligned} E''_A &= L_A + r\ln(L_A+X)\cdot C_A, \text{ and} \\ E''_B &= L_B + r\ln(L_B+X)\cdot C_B. \end{aligned} \tag{2.3}$$

Thus, for the transition to happen, we need

$$f = \frac{\prod_{(i,j)\in A}(L_{ij}+X)^{C_{ij}}}{\prod_{(i,j)\in B}(L_{ij}+X)^{C_{ij}}} > e^{(L_B-L_A)/r.} \tag{2.4}$$

For any $C_B < C_A$, the left-hand side (LHS) of (2.4) is an increasing function with respect to $X > 1$. For the right-hand side (RHS) of (2.4), it is independent of X. Thus, we can control the phase transition by changing X. If we want the phase transition to happen rapidly (at a small value of r), we choose a large X. On the other hand, if we want to smooth the phase transition, we pick a small X. The price we pay for a small X is that we need to increase the value of r to finally reach the C-dominated phase since the RHS of (2.4) is decaying with respect to r. When we compare E, E', and E'', we see that the weight of L in the energy function plays an important



(a) $r = 5.0$. (b) $r = 5.2$. (c) $r = 5.4$.

Figure 2.13: LKH tours with the average distance measure and $E'' = L + r\ln(L + 2.3) \cdot C$ for various values of r.

role in the existence of the phase transition. If L plays too large a role, we have the case of E', i.e., the phase transition does not exist. If L plays too small a role, we have the case of E, i.e., the phase transition exists, but there exists a sharp jump in the transition.

For the numerical experiment, we pick $X \in \{2.3, 2.5, 2.7, 2.9, 3.1\}$ since there is a full phase transition from the L-dominated phase to the C-dominated for all X in the set, keeping the range of r fixed. In Figure 2.12b, as X grows, we see that the phase transition takes place earlier in r and moves to the C-dominated phase more rapidly. Compared to our original energy function E, we also see many more tours with length between 10 and 30, especially when X is small. In Figure 2.13, we observe that these tours with different values of r share many edges. On average, the TSPC tour of $r = 5$ shares 82.6% of its edges with the TSPC tour of $r = 5.4$ when the objective function is E''. This is quite valuable when we try to vary the patrol route frequently without increasing L substantially or completely changing the shape of the tour.

2.2.4 Triangular Path

We have observed substantial crisscrossing when r is large for all three metrics. These crisscrosses are intended to minimize C at the expense of larger L. However, it is undesirable

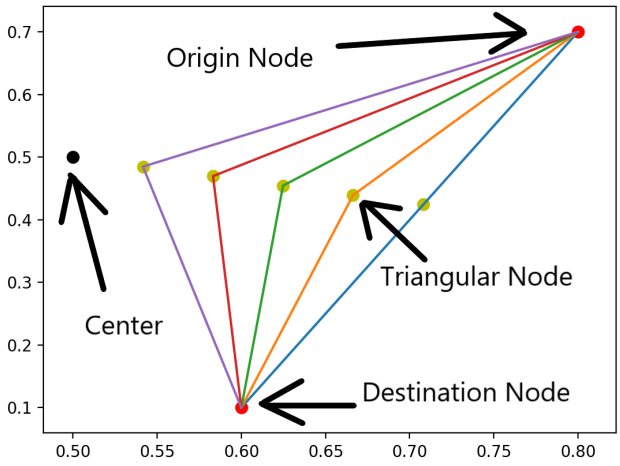


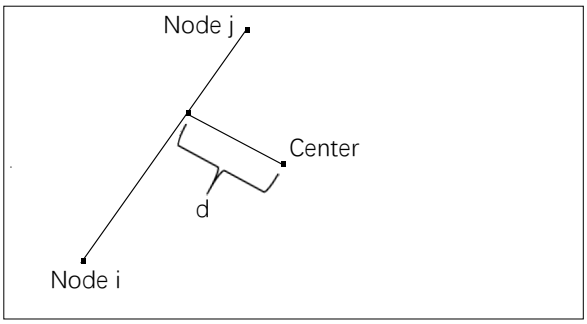
Figure 2.14: An example of triangular paths with 5 different radii.

for the driver to patrol an area using a route with extensive crisscrossing. Moreover, we would like more points in an edge to be close to the center with the average distance measure. For these reasons, we design so-called triangular paths (see Figure 2.14). These paths travel toward the center from an origin node, reach a turning point near the center which we denote as a triangular node, then turn and continue to the next destination node. In order to determine a triangular node, we follow the procedure below.

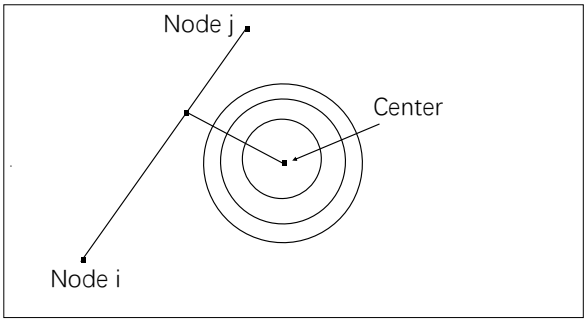
1. Compute the shortest distance d from an edge (i, j) to the center. Figure 2.15a illustrates this step.

2. Create circles centered at the geometric center with radius to be a fraction of d (see Figure 2.15b).

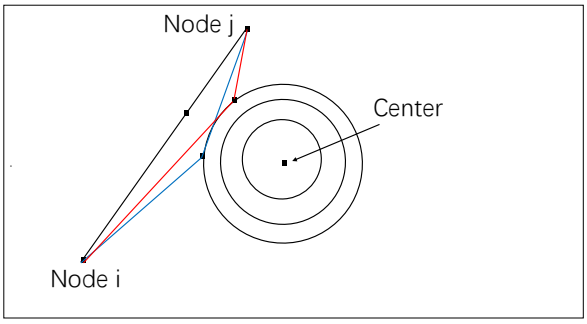
3. For a circle created in Step 2, we sample equispaced points on the circle and find the one with the smallest sum of lengths to nodes i and j. This point is our triangular node. Note that we sample $m = 2500$ points on each circle for the numerical experiment. Figure 2.15c



(a) Compute the minimum distance d between edge (i, j) and the center.



(b) Create circles with radius to be a fraction of d.



(c) Compare the sum of distances from sampled points on the circle to nodes i and j. Note that we are comparing the length of the line segments in blue with those in red here.

Figure 2.15: Steps to find the ideal triangular paths between nodes i and j.

gives an example of this step for two points on the circle.

4. Repeat Step 3 with different radii in order to create other triangular nodes. When the radius is d, the triangular path is the direct edge connecting nodes i and j since the triangular node is on edge (i, j).

5. Connect the triangular nodes to nodes i and j to form the triangular path.

In order to distinguish a tour containing at least one triangular path from a tour containing direct paths only, we call a tour containing at least one triangular path a triangular path tour whereas a tour containing direct paths only is called a direct path tour. It is also important to observe that the procedure above also applies to E'' since E'' is positively correlated with L. Recall that the input for the LKH is a distance matrix D. In the case of the triangular path, it is not immediately clear how one should apply the LKH since there are multiple paths connecting any two nodes. Nevertheless, we can pre-compute the energy function for all triangular paths between nodes i and j and select the smallest value. We further choose this triangular path as our connection between nodes i and j and feed the corresponding E_{ij} or E''_{ij} as our D_{ij}. It is possible that the LKH solution might be slightly suboptimal. As an alternative, we could formulate the problem as an integer program (see below). If there is no constraint on L and an edge from node i to j is in both the solution of the LKH and that of the integer program, then the pre-computed E_{ij} or E''_{ij} is the correct distance. This is not necessarily the case when a constraint on L exists, which will be studied in Section 2.3 below.

Now, let us discuss an IP formulation for the TSPC problem. We denote the set of n destination nodes to be $\mathcal{N} = \{1, 2, \ldots, n\}$ and the number of triangular paths between a pair of nodes to be K. Then, x_{ij}^k is our binary decision variable. In particular, $x_{ij}^k = 1$ if the vehicle

travels from node i to j using the kth triangular path, with $k = 1, ..., K$. Moreover, we have a distance matrix $\mathcal{L}^k$, where l_{ij}^k gives the distance of the route going from node i to j through the kth triangular path. Similarly, we have a distance matrix $\mathcal{C}^k$. For the case of E, c_{ij}^k gives the distance from the kth triangular path between nodes i and j to the center. For the case of E'', c_{ij}^k is the product between the distance from the kth triangular path between nodes i and j to the center and $\ln(X + l_{ij}^k)$. We denote this IP as IP2.1 and present details on IP2.1 below:

$$\text{IP2.1:} \quad \min \qquad \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \sum_{k=1}^{K} \left(l_{ij}^k x_{ij}^k + r c_{ij}^k x_{ij}^k\right) \tag{2.5}$$

$$\text{subject to} \qquad \sum_{j\in\mathcal{N}} \sum_{k=1}^{K} x_{ij}^k = 1, \quad i \in \mathcal{N}, \tag{2.6}$$

$$\sum_{j\in\mathcal{N}} \sum_{k=1}^{K} x_{ji}^k = 1, \quad i \in \mathcal{N}, \tag{2.7}$$

$$\sum_{i\in S} \sum_{j\in S} \sum_{k=1}^{K} x_{ij}^k \leq (|S| - 1), \quad S \subset \mathcal{N}, \tag{2.8}$$

$$x_{ij}^k \in \{0, 1\}, \quad i \in \mathcal{N}, \ \ j \in \mathcal{N}, \ \ k \in \{1, 2, ..., K\}. \tag{2.9}$$

The objective function (2.5) minimizes E or E''. Equations (2.6) and (2.7) ensure every destination node is visited exactly once. The constraints in (2.8) eliminate subtours. Finally, (2.9) describes the domain of our decision variables.

We plot the phase transition of E generated by the two methods (the LKH and IP2.1) over 10 random instances in Figure 2.16a. We find that the LKH tour and the IP2.1 tour are almost identical for every value of r with our test instances. In fact, the relative difference between the length of the LKH tour and the length of the IP2.1 tour is at most 7.2×10^{-3} for our tested instances. Moreover, there exists a jump in L around $r = 4$ for the triangular path. When we

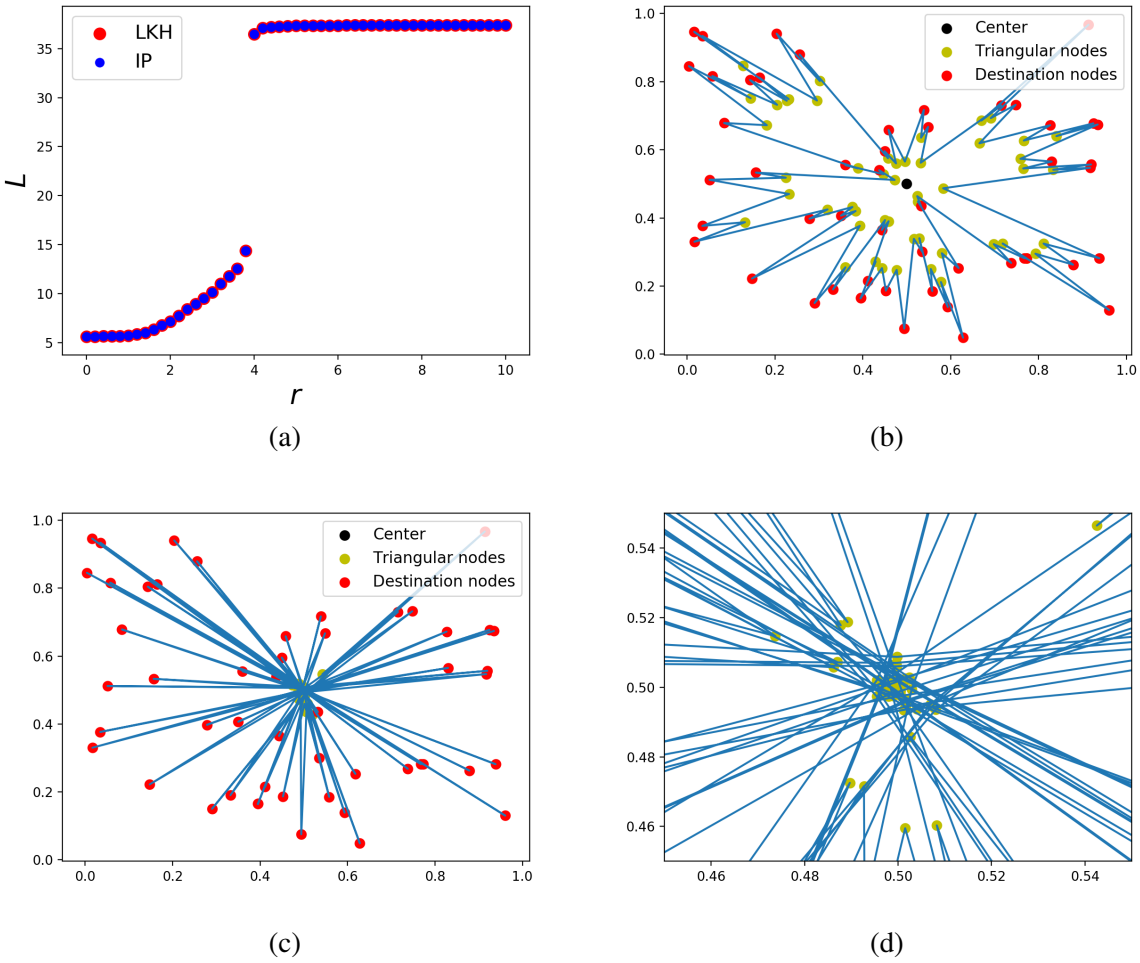
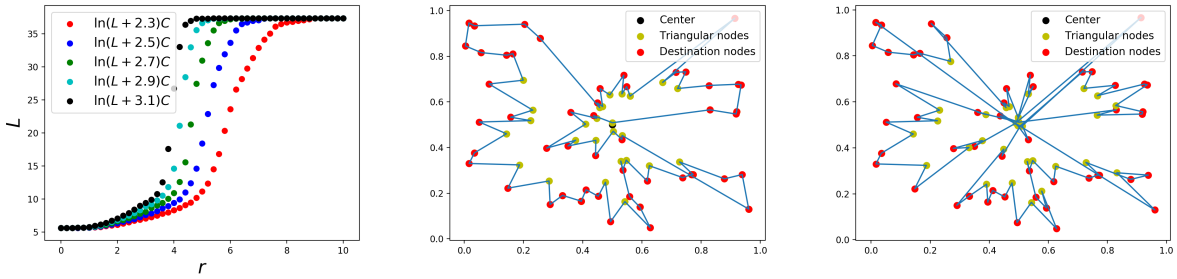


Figure 2.16: Minimize E with the triangular path. Panel (a) plots the phase transition computed by the LKH and IP2.1. Panels (b) and (c) present the triangular path tours when $r = 3.8$ and $r = 4$, respectively, computed by IP2.1. Panel (d) is an enlargement of the center of panel (c).

plot the triangular path tour before the jump in Figure 2.16b, we observe that the tour no longer crisscrosses. Instead, paths in the optimal tour get attracted to triangular nodes near the center. When $r = 4$, in Figure 2.16c, almost all edges get pulled to the triangular nodes closest to the center. Figure 2.16d shows this behavior as an enlargement of the center of Figure 2.16c. In order to verify that E'' also smooths the transition for the triangular path, we plot the phase transition of E'' in Figure 2.17a. As expected, E'' successfully gives us more tours with L between 15 and 35 and brings more direct edges into the tour, as we see in Figures 2.17b and 2.17c.



(a) (b) (c)

Figure 2.17: Minimize E'' with the triangular path: Panel (a) describes the phase transition computed by the LKH for different values of X. Panels (b) and (c) present the triangular path tours of $r = 4$ and $r = 5$, respectively, when $X = 2.3$.

2.3 The Constrained TSPC Problem

We have been focusing on smoothing the phase transition and obtaining intermediate tours during the transition using the average distance measure and E''. Nevertheless, a real-world patrol route usually has an upper bound on the travel distance L and it requires frequent route modifications to avoid premeditated crimes. In our current setup, if we are given an upper bound for L, many runs of the LKH are required by adjusting r in order to find the near optimal tour with the desired travel length. Moreover, this process could be computationally expensive depending on the tolerance of the gap between the length of the LKH tour and the upper bound. Instead, we can solve it by adding a constraint on L to our IP2.1. As a result, only one run of the IP is necessary and the resulting tour is optimal while respecting to the upper bound on L. We denote the IP formulation of our constrained TSPC problem as IP2.2 and provide the details below.

$$\text{IP2.2: } \min \quad \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k=1}^{K}\left(l_{ij}^{k}x_{ij}^{k}+rc_{ij}^{k}x_{ij}^{k}\right) \tag{2.10}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{N}}\sum_{k=1}^{K}x_{ij}^{k}=1, \quad i\in\mathcal{N}, \tag{2.11}$$

$$\sum_{j\in\mathcal{N}}\sum_{k=1}^{K}x_{ji}^{k}=1, \quad i\in\mathcal{N}, \tag{2.12}$$

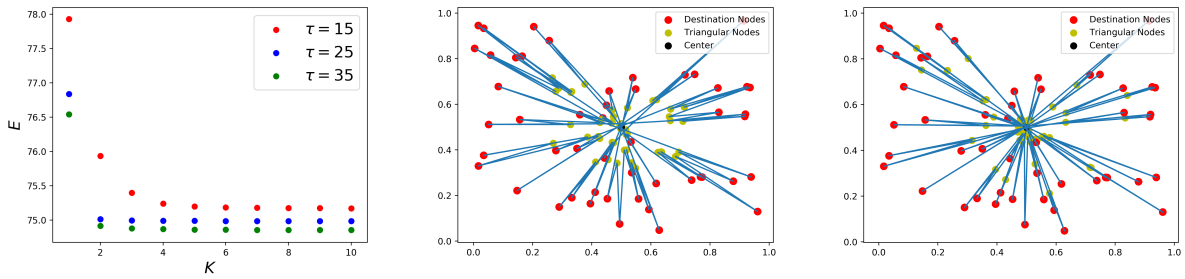
$$\sum_{i\in S}\sum_{j\in S}\sum_{k=1}^{K}x_{ij}^{k}\leq\left(|S|-1\right), \quad S\subset\mathcal{N}, \tag{2.13}$$

$$\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k=1}^{K}l_{ij}^{k}x_{ij}^{k}\leq\tau, \tag{2.14}$$

$$x_{ij}^{k}\in\{0,1\}, \quad i\in\mathcal{N}, \ \ j\in\mathcal{N}, \ \ k\in\{1,2,...,K\}. \tag{2.15}$$

In (2.14), we give the upper bound τ for L. In order to generate τ, we solve the unconstrained TSPC problem. For $r=0$, it is equivalent to solving the TSP, so the resulting tour has the shortest length. Moreover, L is non-decreasing as r increases for TSPC tours. Then, the TSPC tour for $r=10$ has the longest length for our range of r ($0\leq r\leq 10$). We can prove this by contradiction. Let A be the TSPC tour for $r=10$. Assume it does not have the longest length, then there exists some tour B with a strictly longer length and tour B is the TSPC tour for some $r^{*}<10$. By assumption, we have $L_A<L_B$. It follows that $C_A>C_B$. Otherwise, tour A is also the optimal tour when $r=r^{*}$. Moreover, since A is the optimal tour for $r=10$ and B is the optimal tour for $r=r^{*}$, we get that

$$\begin{aligned} &L_A+10C_A<L_B+10C_B, \text{ and} \\ &L_A+r^{*}C_A>L_B+r^{*}C_B. \end{aligned} \tag{2.16}$$



(a) (b) (c)

Figure 2.18: Solve IP2.2 with different number of triangular paths per edge: Panel (a) presents the relationship between E and K for three different bounds τ on L. Panel (b) gives the constrained TSPC tour when $K = 2$ and $\tau = 25$. Note that $L = 24.96$ and $C = 12.36$ in (b). Panel (c) shows the constrained TSPC tour when $K = 5$ and $\tau = 25$. Note that $L = 24.99$ and $C = 12.35$ in (c).

Rearranging (2.16), we have

$$10(C_A - C_B) < (L_B - L_A) < r^*(C_A - C_B). \tag{2.17}$$

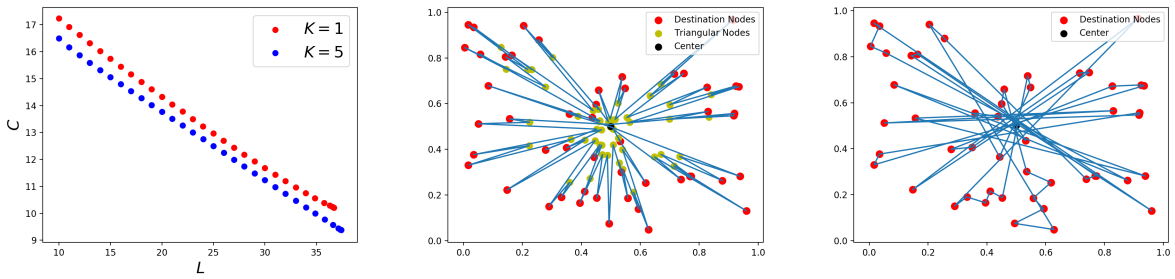
Since $r^* < 10$ and $C_A > C_B$, this is a contradiction and it implies that the length of the TSPC tour for $r < 10$ must be smaller than or equal to that of the TSPC tour for $r = 10$. Thus, we consider the length L^{lo} of the tour of the unconstrained TSPC problem with $r = 0$ as a lower bound of τ and the length L^{hi} of the tour of the unconstrained TSPC problem with $r = 10$ as an upper bound of τ.

IP2.2 not only helps us find the tour with the desired length but also gives us a tool to analyze the relationship between E and the number of triangular paths K. In other words, we can figure out how much we gain from adding more triangular paths while keeping L fixed. For the numerical experiment, we vary K from 1 to 10. In particular, the ith triangular path between a pair of nodes has the triangular node with id/K distance away from the center for $1 \leq i \leq K$, where d is the shortest distance between the direct edge connecting the nodes and the center.

Note that when $K = 1$, we have the direct path tour. We then minimize E with IP2.2 when $r = 4$ (the location of the jump, as shown in Figure 2.16a) and $\tau \in [15, 25, 35]$. In Figure 2.18a, we plot the average of E over 10 instances versus K. We observe that $\tau = 15$ (corresponds to the L-dominated phase) benefits from a larger K, i.e., E has decreased continuously from $K = 1$ to $K = 5$. On the other hand, for $\tau = 25$ or $\tau = 35$, there is a large change in E when we shift from the direct path tour to the triangular path tour ($K = 1$ to $K = 2$). Then, the change in E flattens. We interpret this phenomenon as reflecting the fact that there is more freedom to choose edges in the L-dominated phase than in the C-dominated phase. It is common that in the C-dominated phase, almost all edges get pulled to the triangular node nearest to the center, as we see in Figure 2.18b and Figure 2.18c. Even when $K = 2$, we are able to find a tour with L close to τ and a competitive C.

One common behavior, observed in Figure 2.18a, is that the triangular path tour beats the direct path in E no matter what the bound τ is. This suggests that when solving IP2.2 with the same τ, the triangular path tour may be closer to the center in the average distance measure than the direct path tour. In order to verify this observation, we simplify IP2.2 and change the objective function to minimize C to get rid of the r dependence. In particular, (2.10) in IP2.2 becomes $\sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \sum_{k=1}^{K} c_{ij}^k x_{ij}^k$. Removing the r dependence gives us a fair comparison because the direct path tour and the triangular path tour may be at different phases of the transition for the same value of r.

We pick τ ranging from 10 to 40 in increments of 1 for the numerical experiment and solve IP2.2 by the direct path $(K = 1)$ and the triangular path $(K = 5)$ over 10 random instances. For the resulting IP2.2 tours, we then plot the distance to the center C versus the length L in Figure 2.19a. Note that for each τ, C and L are averaged over the tested instances. We see that the



(a) (b) (c)

Figure 2.19: Solve IP2.2 with the direct path ($K = 1$) and the triangular path ($K = 5$): Panel (a) shows C as a function of L for the direct path and the triangular path. Panels (b) and (c) present the constrained TSPC tours for the triangular path and the direct path, respectively, when $\tau = 20$.

triangular path tour has a smaller C than the direct path tour when imposing the same constraint τ on the length. The triangular path tour also offers tours with a larger range in C than the direct path method. In particular, one would choose the triangular path tour when seeking to minimize C without paying attention to L. Moreover, the triangular path tour in Figure 2.19b is a more sensible tour since it connects nodes close to each other by the triangular path. On the other hand, the direct path tour in Figure 2.19c favors crisscrossing even with the average distance measure.

2.4 Runtime Analysis

We have shown that the triangular path has an advantage over the direct path with respect to minimizing C, while total tour length, L, is constrained. One may also be interested in comparing the runtimes of the direct path and the triangular path approaches. For this purpose, we compute the runtime for IP2.2 when minimizing C with $K = 1$ and $K = 5$. We note that the runtime is split into two parts: the time spent pre-computing the matrices $\mathcal{L}^k$ and $\mathcal{C}^k$ (pre-runtime) and the time spent solving IP2.2 (IP time). For the bound τ, we consider three values: $1.5L^{lo}$, $4L^{lo}$, and $7L^{lo}$, where L^{lo} is again the length of the tour for the TSPC with $r = 0$. These three τ values

n	K	Pre-runtime	IP time (Sharp)	std	IP time (Mild)	std	IP time (Loose)	std
20	1	0.06	0.16	0.14	0.04	0.02	0.03	0.02
20	5	1.60	0.28	0.05	0.21	0.21	0.13	0.03
30	1	0.14	0.32	0.13	0.38	0.71	0.06	0.01
30	5	3.74	0.73	0.11	1.20	0.80	0.51	0.22
40	1	0.24	0.99	0.47	1.73	2.05	0.12	0.03
40	5	6.37	1.50	0.31	1.89	0.60	0.91	0.24
50	1	0.40	1.49	0.71	2.84	1.32	0.28	0.15
50	5	10.04	2.58	0.57	3.74	0.59	2.65	1.69
60	1	0.55	2.84	0.99	4.56	2.38	3.53	4.00
60	5	15.37	4.55	1.28	5.84	0.78	5.95	2.44
70	1	0.75	4.02	2.05	10.36	8.93	13.00	9.18
70	5	19.76	6.89	1.81	9.99	1.97	9.81	2.80
80	1	0.96	5.50	1.59	14.32	8.43	21.88	19.09
80	5	27.06	11.54	3.21	13.99	1.50	15.62	1.54

Table 2.2: Runtime analysis for IP2.2 with different levels of bounds on L and two values of K. Note that the time recorded is in seconds.

correspond to a sharp bound, a mild bound, and a loose bound, respectively, on the length of a tour L. Moreover, by varying the number of nodes n from 20 to 80 in increments of 10, we obtain the mean of the pre-runtime and the IP time and the standard deviation (std) of the IP time over 10 instances, which we record in Table 2.2. Note that IP2.2 is solved by Gurobi 9.1.2 via Python API and run on a 3.2 GHz 8-core Apple M1 with 8 GB of RAM using macOS.

In Table 2.2, the pre-runtime for the direct path is much shorter than it is for the triangular path since the computational complexity is $O(n^2p)$ for the direct path and $O(n^2pm)$ for the triangular path in the pre-computing stage, where $(p-1)$ is the number of sampled points on edge (i,j) when computing C_{ij}^a and m is the number of sampled points on each circle when computing the triangular nodes. Nevertheless, we only need to run the pre-computing stage once

since the matrices $\mathcal{L}^k$ and $\mathcal{C}^k$ are independent of τ. Thus, the longer pre-runtime for the triangular path is not that troublesome for our range of n.

For the IP time, the direct path beats the triangular path for the sharp bound case. This advantage grows as n increases. Note that most edges in the optimal tour are direct in the sharp bound case. Therefore, for the triangular path, having multiple paths between a pair of nodes is unnecessary and also a computational burden for our IP2.2. For the mild bound, the triangular path has slightly larger IP times than the direct path when $n \leq 60$. However, the triangular path outperforms the direct path with respect to IP times and the standard deviation when $n \geq 70$. Moreover, using t-test with a significance level of 5%, we observe that the significant difference only exists for $n = 20$ and $n = 30$. Note that we use the callback function for the subtour elimination constraint. Specifically, we first solve a relaxed version of the formulation without any subtour elimination constraint. Then, we check if the resulting tour has any subtour. If there is any, we break the subtour by imposing the subtour elimination constraints. We then solve the resulting formulation again with the newly included subtour elimination constraints. We keep iterating the process until the solution has no subtour. Thus, the runtime also depends on how fast we can find a potential solution without subtours. In the case of the triangular path, there is a broader range of edges for IP2.2 to choose from to satisfy the constraint on L. For the loose bound, the direct path beats the triangular path in IP time when $n \leq 60$. However, as $n \geq 70$, the direct path's IP times deteriorate and fluctuate among instances. On the other hand, the triangular path has a smaller and more stable runtime.

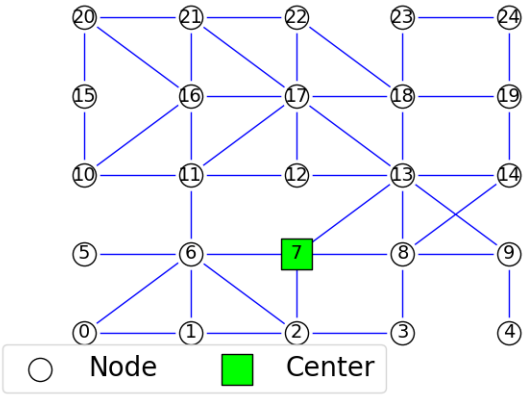


Figure 2.20: An example of our grid network.

2.5 Extending the TSPC to Grid Networks

In our current setup, we assume a Euclidean graph with the Euclidean distance as our measure of distance. This assumption works well for patrol conducted by helicopters, drones, and boats. However, in the case of ground-based police patrol, a simple, undirected, and connected grid network is more appropriate and realistic. In our unit square grid, we set the grid size in the horizontal direction to be h_H and the grid size in the vertical direction to be h_V. A node is located at the coordinates (ih_H, jh_V) for $0 \leq i \leq 1/h_H$ and $0 \leq j \leq 1/h_V$. There are possible edges connecting adjacent nodes in the horizontal, vertical, and diagonal directions. The probability of a horizontal or vertical edge is 0.9 whereas the probability of a diagonal edge is 0.5. We also generalize the definition of the center such that the center can be any node in the grid and does not need to be fixed at the geometric center. Since the center is one of the nodes in the grid network, it will be visited in the tour. In Figure 2.20, we plot an instance of our grid network.

2.5.1 Defining the TSPC in Grid Networks

In order to model the TSPC over a grid network, we need to redefine L and C. In particular, the measure of distance changes to the shortest path distance over the grid network. Let S be the set of shortest paths between nodes i and j. If $|S| = 1$, L_{ij} and C_{ij} are the length and the average distance to the center of the shortest path in S, respectively. Otherwise, since the shortest paths in S share the same length, we can still use L_{ij} for their lengths. However, they could have different average distances to the center. For $s \in S$, C_{ij}^s measures the average distance from s to the center and $C_{ij} = \min_{s \in S} C_{ij}^s$.

To find C_{ij}^s, one computes the average distance from each edge in s to the center. Let (k, l) be an edge in s. For a point p on edge (k, l), we are interested in the shortest path distance from point p to the center. As a result, if there exists more than one shortest path between point p and the center, we only need to consider one of them because the shortest paths share the same length. Observe that a shortest path from point p to the center must reach node k or l first, then follow a shortest path from node k or l to the center. The choice between nodes k and l depends on the location of point p on the edge and the size of L_{kc^*} and L_{lc^*}, where c^* is the index of the center. For instance, if $0 \leq L_{kc^*} - L_{lc^*} \leq L_{kl}$, then the point p^* on the edge with the distance $(L_{kl} - (L_{kc^*} - L_{lc^*}))/2$ to node k is the critical point. In other words, for points on the edge between node k and point p^*, the shortest paths travel to the center via node k. For points on the edge between node l and point p^*, the shortest paths travel to the center via node l. Observe that when these shortest paths reach node k or l, the distance to the center can only be L_{kc^*} or L_{lc^*}. Thus, the computation of the average distance from edge (k, l) to the center can be reduced to two problems. First, we obtain the percentage of points on the edge for which

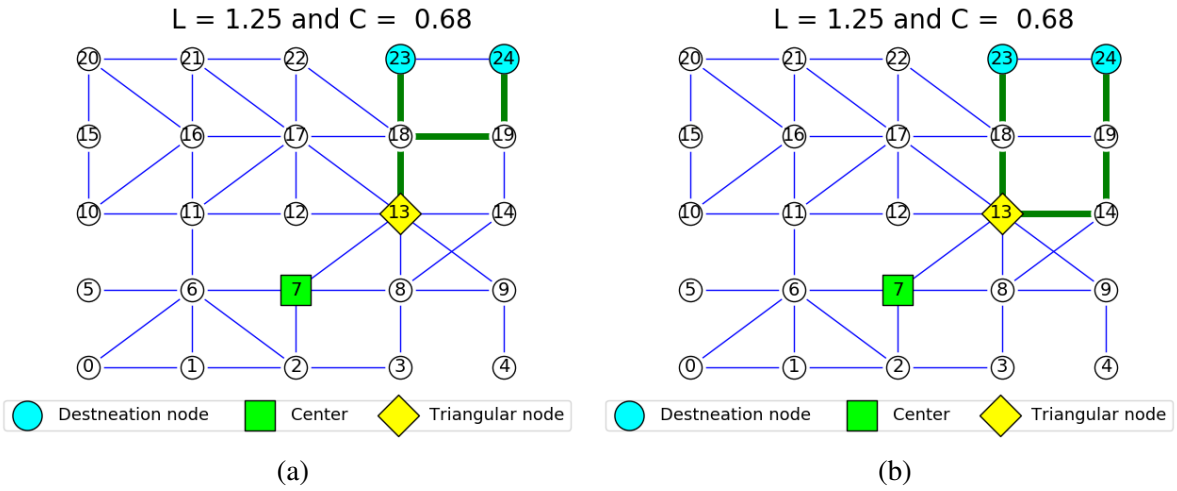
the shortest paths to the center pass through node $k(l)$. Second, for the points with node $k(l)$ in the shortest paths to the center, we compute their average distance to node $k(l)$. Since these two problems can be solved exactly, the average distance from edge (k,l) to the center is exact. Let $d_{kl} = (L_{kl} - |L_{kc^*} - L_{lc^*}|)/2,\ mx_{kl} = \max(L_{kc^*}, L_{lc^*}),\ mi_{kl} = \min(L_{kc^*}, L_{lc^*})$. Then, the average distance to the center of edge (k,l) can be defined as

$$g_{kl} = \begin{cases} mi_{kl} + L_{kl}/2, & \text{if } |L_{kc^*} - L_{lc^*}| \geq L_{kl}, \\ (d_{kl}(mx_{kl} + d_{kl}/2) + (L_{kl} - d_{kl})(mi_{kl} + (L_{kl} - d_{kl})/2))/L_{kl}, & \text{if } |L_{kc^*} - L_{lc^*}| < L_{kl}. \end{cases} \tag{2.18}$$

Consequently, $C_{ij} = \min_{s \in S} C_{ij}^s = (1/L_{ij}) \min_{s \in S} \sum_{(k,l) \in s} g_{kl} L_{kl}$. From now on, when we mention the shortest path between nodes i and j, we are referring to the shortest path(s) with the smallest average distance to the center.

2.5.2 Triangular Path in Grid Networks

We have seen that the triangular path method helps us get closer to the center in a Euclidean graph. In order to reconstruct a triangular path in a grid network, we need to specify the location of a triangular node. One approach is to consider a triangular node made up of existing nodes. Hence, when commuting from node i to j via triangular node t, the vehicle traverses the shortest path between nodes i and triangular node t and the shortest path between triangular node t and node j (see Figure 2.21b). When $t = i$ or j, the triangular path is just the shortest path between nodes i and j, which we will refer to as the direct path. To quantify the quality of a triangular path between nodes i and j, we compute $L_{it} + L_{tj}$ and $(C_{it}L_{it} + C_{tj}L_{tj})/(L_{it} + L_{tj})$ for $t \in \mathcal{N}$,



(a) (b)

Figure 2.21: Two triangular paths with identical triangular nodes, length, and average distance to the center. In panel (a), the sequence of the triangular path is [23,18,13,18,19,24]. In panel (b), the sequence of the triangular path is [23,18,13,14,19,24].

i.e., the length of the triangular path and the average distance to the center of the triangular path. In order to narrow down the possibilities for a triangular path between nodes i and j, we use the following restrictions:

$$L_{it} + L_{tj} \leq L_{ic^*} + L_{c^*j} \text{ and} \tag{2.19a}$$

$$(C_{it}L_{it} + C_{tj}L_{tj})/(L_{it} + L_{tj}) \leq C_{ij}. \tag{2.19b}$$

In (2.19a), we force the length of a triangular path to be shorter than the length of the triangular path with the center as the triangular node (the center triangular path). This ensures that among the qualified triangular paths between nodes i and j, the center triangular path has the longest length and the shortest average distance to the center. In (2.19b), we keep the average distance to the center of a triangular path shorter than the average distance to the center of the direct path. In other words, we do not include triangular paths that travel a longer distance and are farther from

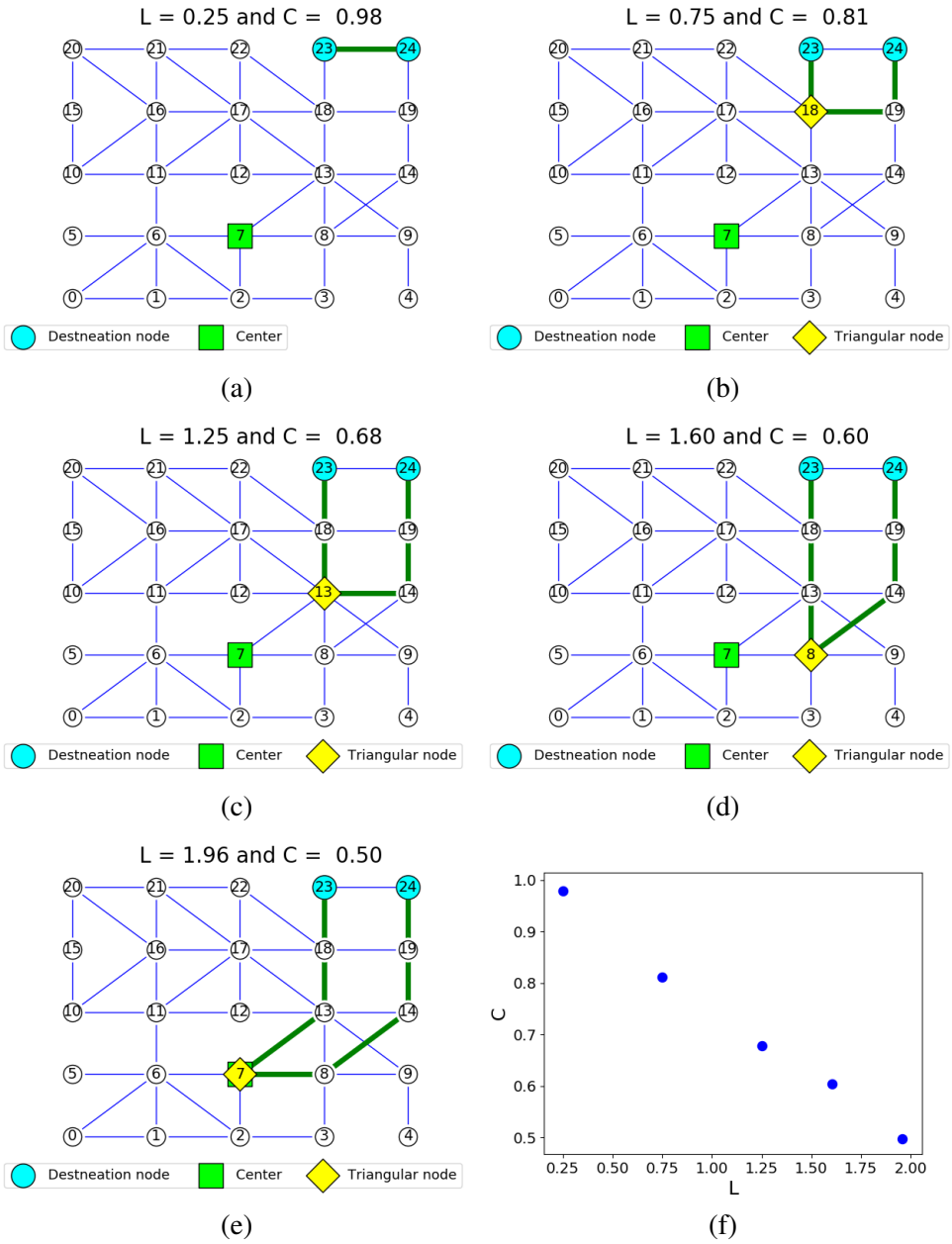
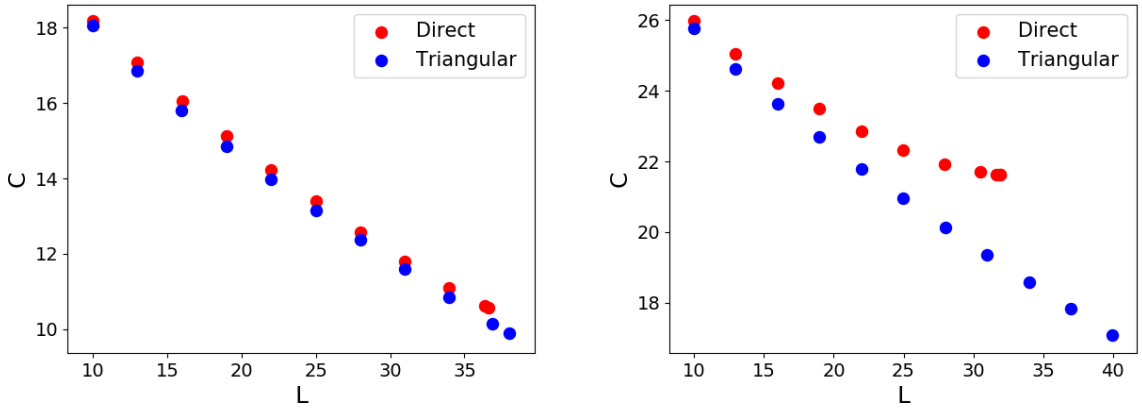


Figure 2.22: Triangular paths in a grid network. In panels (a) - (e), we plot the triangular paths between nodes 23 and 24. In panel (f), we plot the length versus the average distance to the center of the triangular paths.

the center than the direct path. Moreover, if there is more than one shortest path connecting node i and the triangular node or node j and the triangular node, then the chosen triangular path should visit the maximum number of nodes. This avoids, as much as possible, a U-turn at the triangular

node and widens the range of visited nodes in the triangular path while maintaining the same average distance to the center. For instance, in Figure 2.21, we plot two triangular paths between nodes 23 and 24. These triangular paths have the same triangular node (node 13). However, the triangular path in Figure 2.21b visits more nodes and hence is chosen.

Once the feasible triangular paths between nodes i and j are identified, we sort them according to their average distances to the center. In particular, the direct path has the longest average distance to the center, and the center triangular path has the shortest average distance to the center. Recall that the number of triangular paths between a pair of nodes is K for the Euclidean graph case. For the grid network case, the number of feasible triangular paths varies due to (2.19a) and (2.19b). As a result, we select at most K triangular paths from all feasible paths. Let l be the number of feasible triangular paths between nodes i and j. If $l \leq K$, we will keep all feasible triangular paths. Otherwise, we first select two paths: the direct path and the center triangular path. The kth path selected is the triangular path with the $(1+(k-2)\lceil l/K \rceil)$th longest average distance to the center for $3 \leq k \leq K$. For instance, if $l = 9$ and $K = 3$, we select the direct path, the center triangular path, and the triangular path that has the fourth longest average distance to the center. In Figure 2.22, we plot the triangular paths between nodes 23 and 24 when $K = 5$. We can see the trade-off between L and C, which was also observed in the Euclidean graph case. Hence, for the triangular path, there is nothing unique about the assumption of a Euclidean graph. Instead of traveling from a triangular node to a destination node in a straight line, we traverse via the shortest path between them in the grid network.



(a) Center located at the geometric center.

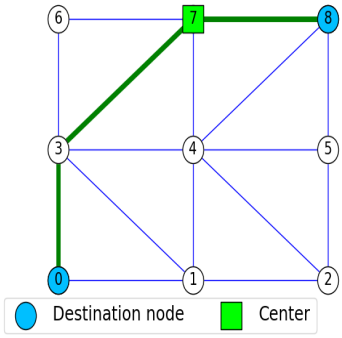
(b) Center located at a random node.

Figure 2.23: Comparison between the direct path tour and the triangular path tour in the grid networks.

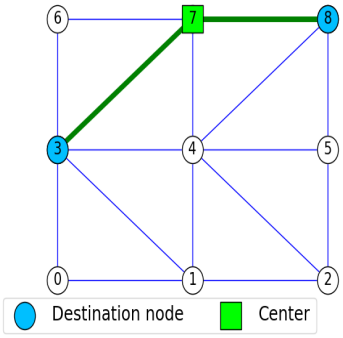
2.5.3 Direct Path vs. Triangular Path in Grid Networks

Now that we have redefined L, C, and the triangular path in a grid network, we will show that our triangular path tour still outperforms the direct path tour. For this purpose, we use IP2.2 to minimize C subject to a constraint τ on L. We pick τ ranging from 10 to 40 and solve IP2.2 by the direct path and the triangular path. For the triangular path, we set the maximum number of triangular paths between a pair of nodes as 5. In Figure 2.23, we plot the relationship between L and C in the resulting IP2.2 tours for the case of the center as the geometric center node and the case of the center as a random node. Note that C and L are averaged over 10 random 7×5 grids. Overall, in both cases, the triangular path outperforms the direct path, i.e., the triangular path tour is closer to the center than the direct path tour when the same constraint τ is imposed. When the center is not at the geometric center, the margin of the triangular path tour's outperformance increases.

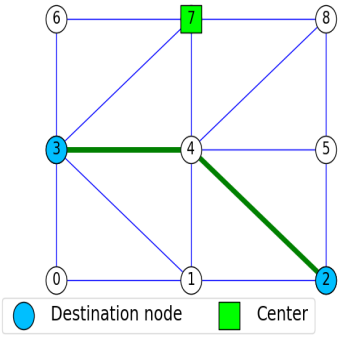
We notice that the triangular path tour and the direct path tour diverge more substantially



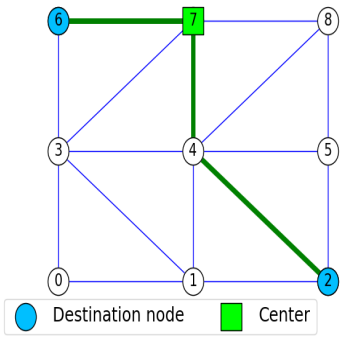
(a) Sequence of the path: $\{0, 3, 7, 8\}$.



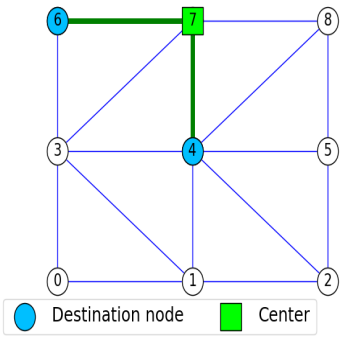
(b) Sequence of the path: $\{8, 7, 3\}$.



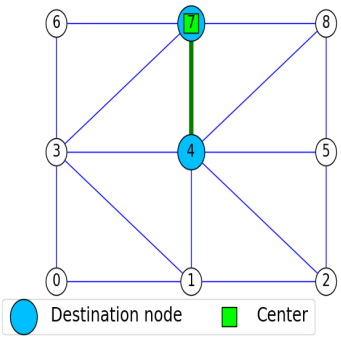
(c) Sequence of the path: $\{3, 4, 2\}$.



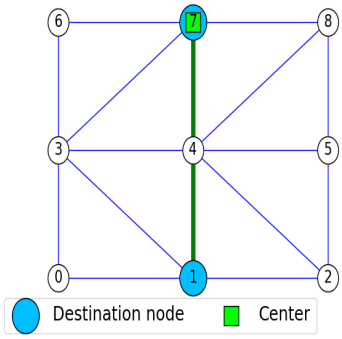
(d) Sequence of the path: $\{2, 4, 7, 6\}$.



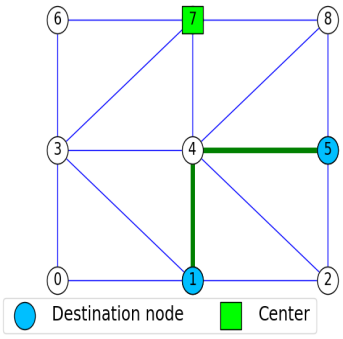
(e) Sequence of the path: $\{6, 7, 4\}$.



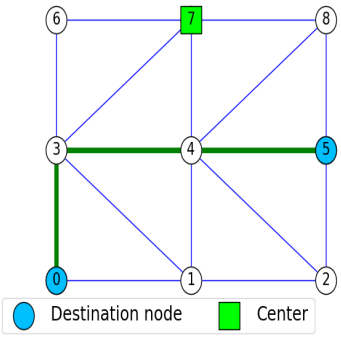
(f) Sequence of the path: $\{4, 7\}$.



(g) Sequence of the path: $\{7, 4, 1\}$.



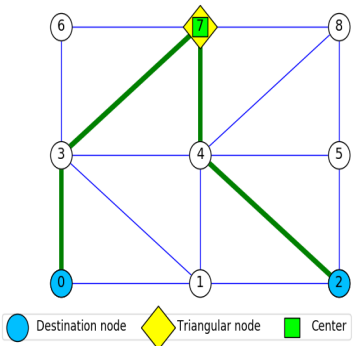
(h) Sequence of the path: $\{1, 4, 5\}$.



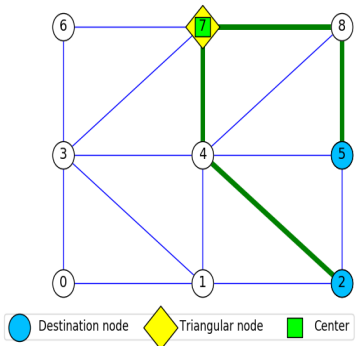
(i) Sequence of the path: $\{5, 4, 3, 0\}$.

Figure 2.24: The direct path tour with the smallest C. Note that $L = 10.83$ and $C = 4.66$ for the direct path tour.

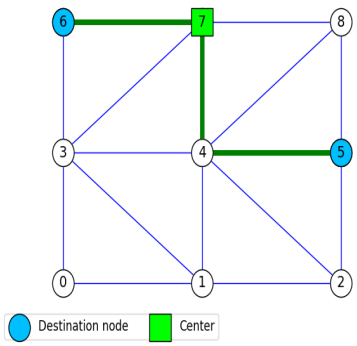
when the center is a random node and the bound on L is loose. To further understand this phenomenon, we minimize C with no constraint on L and plot the resulting direct path tour and



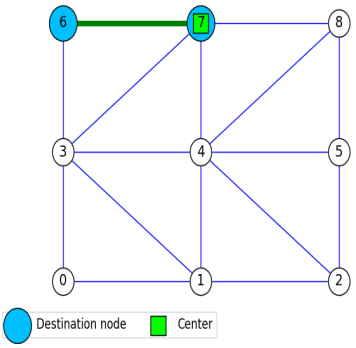
(a) Sequence of the path: $\{0, 3, 7, 4, 2\}$.



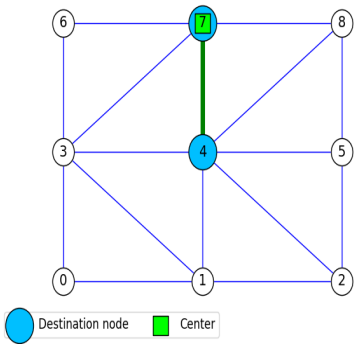
(b) Sequence of the path: $\{2, 4, 7, 8, 5\}$.



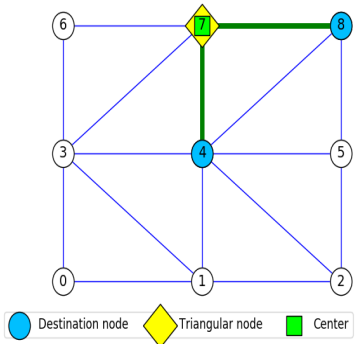
(c) Sequence of the path: $\{5, 4, 7, 6\}$.



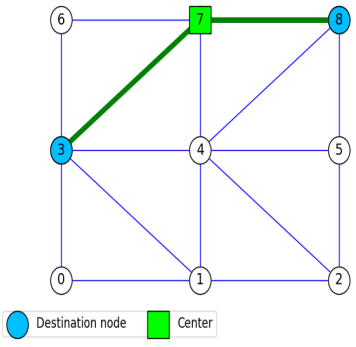
(d) Sequence of the path: $\{6, 7\}$.



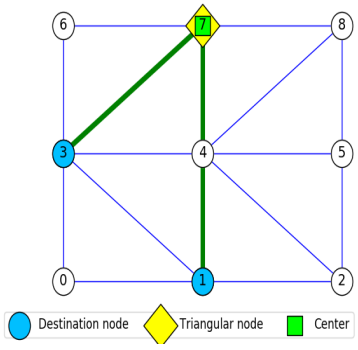
(e) Sequence of the path: $\{7, 4\}$.



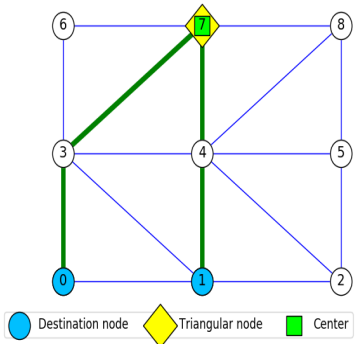
(f) Sequence of the path: $\{4, 7, 8\}$.



(g) Sequence of the path: $\{8, 7, 3\}$.



(h) Sequence of the path: $\{3, 7, 4, 1\}$.



(i) Sequence of the path: $\{1, 4, 7, 3, 0\}$.

Figure 2.25: The triangular path tour with the smallest C. Note that $L = 13.24$ and $C = 3.63$ for the triangular path tour.

triangular path tour in Figures 2.24 and 2.25. This experiment is done in a 3×3 grid. Each panel of Figures 2.24 and 2.25 plots the path between two consecutive nodes in the tour. We observe

that the direct path tour gets closer to the center by connecting two nodes such that the shortest path between them includes the center. However, it is less possible to include the center when the center is in the periphery of the square. As a result, C is much larger for the direct path tour. On the other hand, the triangular path tour offers us the extra option of using the center as the triangular node. Hence, the triangular path tour is insensitive to the location of the center.

2.6 Conclusion

We revisit the midpoint measure in Lipowski and Lipowska [41] and the nearest point measure in Price [48] and illustrate their main weaknesses: these metrics do not give an accurate C_{ij}. To address these weaknesses and provide an alternative, we propose the average distance measure. Moreover, we present a new energy function, E'', which includes a logarithmic function. With the help of E'', we successfully smooth the phase transition and offer a more gentle transition from the L-dominated phase to the C-dominated phase. Then, by defining the triangular path, we obtain a more sensible tour that, unlike the direct path case, does not crisscross. We also show that in both a Euclidean graph and a grid network and under the same constraints on the travel length, the triangular path tour has a shorter distance to the center than the direct path tour. We use the runtime analysis to demonstrate that the runtime for the triangular path is competitive with the runtime for the direct path.

Future work can extend our exploration of the TSPC problem to cases with multiple vehicles or multiple centers. For instance, for the multiple vehicle case, it would be interesting to design vehicle tours such that one of the vehicles is always close to the center. Moreover, research can explore how the constrained TSPC problem can be transformed into a bin-packing

problem. The objective of the bin-packing problem would be to minimize E with L constrained to be no more than $(1 + h)L^{lo}$, where h is a positive constant. Thus, one would choose from the set of triangular paths in order to maximize the reduction in C compared to the TSPC tour with $r = 0$. If runtime deteriorates for the constrained TSPC problem, one could then reduce runtime by switching to the bin-packing problem and applying a heuristic approach (e.g., a greedy algorithm).

Chapter 3: The Hot Spot Coverage Patrol Problem: Formulations and Solution Approaches

3.1 Introduction

A preventive patrol is defined as one in which police or private security officers travel around an area following a set of routes in order to lower the crime rate and increase the confidence of the local population [3, 34]. In preventive patrols, there often exist hot spots, which are locations with high crime rates. These hot spots require regular visits by patrollers because the physical presence of police lowers the inclination of an individual to commit a crime [20]. Designing patrol routes with frequent visits to hot spots is never an easy task. First, in order to cover hot spots efficiently and frequently, continuous communication is required between patrollers [16], which often leads to a dynamic routing problem. Second, although hot spots are very important, patrollers must also spend time performing other tasks such as emergency dispatches [65]. Third, choosing an objective for a patrol route is not trivial. There are several options, such as minimizing the response time to a crime [54, 58], maximizing coverage of the hot spots [12, 14, 17, 22, 31], and minimizing the time lag between two consecutive visits to hot spots [15].

In this chapter, we consider a patrol routing problem for a fleet of patrol cars in an urban

region, which contains a high-crime neighborhood (HCN) with multiple hot spots. Due to the severity and frequency of crimes in the HCN, typically, at least one patrol car is required in the HCN at any given time during the patrol. We call this requirement the constant presence requirement in the HCN. Yet resources, such as patrol time and the number of patrol cars are often limited, and, as a result, it is impossible to visit all of the patrol locations. Therefore, the locations with higher crime or security risks are prioritized in daily operations. We then quantify the importance of a patrol location by assigning an imaginary prize to it, and a patroller collects the prize upon visiting the location. Given a set of patrol locations, travel times, and prizes, the goal is to maximize the sum of prizes collected in which we decide the locations to visit, the route, and the starting point of each patrol car while obeying operations constraints. In addition to the constant presence requirement in the HCN, each patrol car has to respect the maximum work shift, and is required to enter and exit the HCN exactly once to have continuous coverage of the HCN. We refer to this problem as the Hot Spot Coverage Patrol Problem (HSCPP). The solution of the HSCPP describes the patrol routes and the starting points of the patrol cars. Note that the starting point is a variable in the HSCPP because patrollers start their day at the assigned starting point, i.e., visiting the police depot first is unnecessary. This reduces the commuting time and increases the patrol time of a patroller [57].

The HSCPP shares some similarities with the celebrated Vehicle Routing Problem (VRP) [28, 46, 60, 61]. However, the HSCPP needs to decide the starting points for the patrol cars without given depots, as in the VRP. This imposes a new challenge in modeling and solution approaches. The closest problem studied in the literature is the Maximum Coverage and Patrol Routing problem (MCPRP) introduced by Keskin et al. [31]. In the MCPRP, there exist hot spots in the highway where traffic accidents are prone to happen. In addition, each hot spot is only

active within a specific time period, i.e., the coverage of a hot spot is only valid if a patroller is at the hot spot during the period. For a fleet of patrollers, the objective of the MCPRP is to find the optimal patrol routes to maximize coverage of the hot spots while obeying the limited resource constraints. When solving the MCPRP, methods such as mixed integer programming, tabu search, and branch-and-price are applied in Keskin et al. [31], Çapar et al. [12], Dewil et al. [22], and Chircop et al. [17]. There are several key differences between the HSCPP and MCPRP. First, in real life, hot spots are weighted by the frequency of traffic accidents or the crime rate, and in the MCPRP, little attention is paid to distinguishing between hot spots, which is not the case for the HSCPP. Second, in the MCPRP, the starting point of the tour is either the police depot or the patroller's home. In the HSCPP, the starting point is set to be a variable to increase patrol time and satisfy the constant presence requirement in the HCN. Third, the MCPRP deals with highway patrol where hot spots are distributed throughout the region. The HSCPP is more applicable to regions where hot spots are concentrated in some area, such as the city center.

To solve the HSCPP, we first make an important observation regarding the constant presence requirement in the HCN. A necessary and sufficient condition for the requirement is that the sum of time that all patrol cars spend in the HCN is greater than the duration of the patrol. Furthermore, once the routes are determined, the total time spent in the HCN is fixed. Therefore, we can separate the process of finding the routes and starting points into two steps.

For the more challenging first step of finding the routes, we develop several solution approaches and conduct a detailed computational case study based on real data from Montgomery County, Maryland to evaluate the effectiveness of our approaches. The global approach consists of finding the routing solution for all patrol cars with a single integer programming (IP) formulation. We show that the optimality gap of the global approach is 5.5%, on average, while it has

high variance and can be as large as 22.5%, after an hour of runtime. On the other hand, the partition approach involves first partitioning the region geographically and solving the routing problem in each subregion with two IP formulations. The runtimes of the partition approach are around 5 minutes on average, and the optimality gap is within 5%. This promising result justifies the validity of applying the partition approach for the HSCPP. Next, we strengthen the partition approach by developing a column generation (CG) approach, where the initial columns of the CG approach are the solutions generated from the partition approach. The optimality gap of the CG approach averages 2.4%, with a maximum of 6.7% after around 36 minutes of runtime. In addition, by using the best solution from the partition approach as a warm start for the global approach, the optimality gap can be further reduced to 2.1%, on average, with a maximum of 5.4%, after 55 minutes of runtime.

For the second step of finding the start points, we have a polynomial-time procedure. To achieve this, we leverage the tours found in the first step and implement a simple search algorithm with computational complexity $O(KN)$, where K is the number of patrol cars and N is the number of patrol locations.

Workload balancing has received more attention recently [43]. We consider workload balancing in the HCN because the crime rate is higher there. It is better to spread the HCN patrol among all patrollers, instead of letting one or two patrollers focus on the HCN. To quantify workload balancing, we use the standard deviation of the time spent in the HCN among patrollers. Ideally, we prefer the standard deviation to be small. Overall, we recommend using our partition approach if workload balancing and runtime are the priorities. On the flip side, we recommend using the global approach with a warm start because this approach finds a nice balance between the optimality gap and the workload balancing.

The remainder of this chapter is structured as follows. In Section 3.2, we formally define the HSCPP. In Section 3.3, we present the global approach, partition approach, and the CG approach to find the routes for the HSCPP. In Section 3.4, a search algorithm is presented to find the starting points of the tours found in Section 3.3. In Section 3.5, a detailed computational case study based on real data in Montgomery County, Maryland is conducted to evaluate the effectiveness of our solution approaches. Some concluding remarks are offered in Section 3.6. Finally, the technical proofs are included in Appendix A

3.2 Problem Setting

In this section, we formally define the HSCPP and relevant terms and concepts such as HCN, patrol speed, duration of the patrol, and idle time.

In the HSCPP, the patrol network is a directed and complete network $G = (V, A)$, where V is the set of nodes describing patrol locations and A is the set of arcs connecting patrol locations. For the distance metric, the distance between nodes i and j is denoted by L_{ij}. The fleet of K homogeneous patrol cars is denoted by $\mathcal{K}$. In addition, we have a time limit of T hours for a patrol route. That means each patrol route is T-hours long and the coverage duration in the HCN is T hours. During this coverage duration, at least one patrol car is in the HCN. Also, we have a distance limit of τ miles for a patrol route. These two limits are linked by the patrol speed v as $Tv = \tau$.

For ease of illustration, we model the HCN as a circular region as shown in Figure 3.1. The exact location of the HCN is determined by plotting the heat map of crimes in a region based on historical data. For the set of nodes in the HCN, we denote the set of nodes by V_{inner}. They are

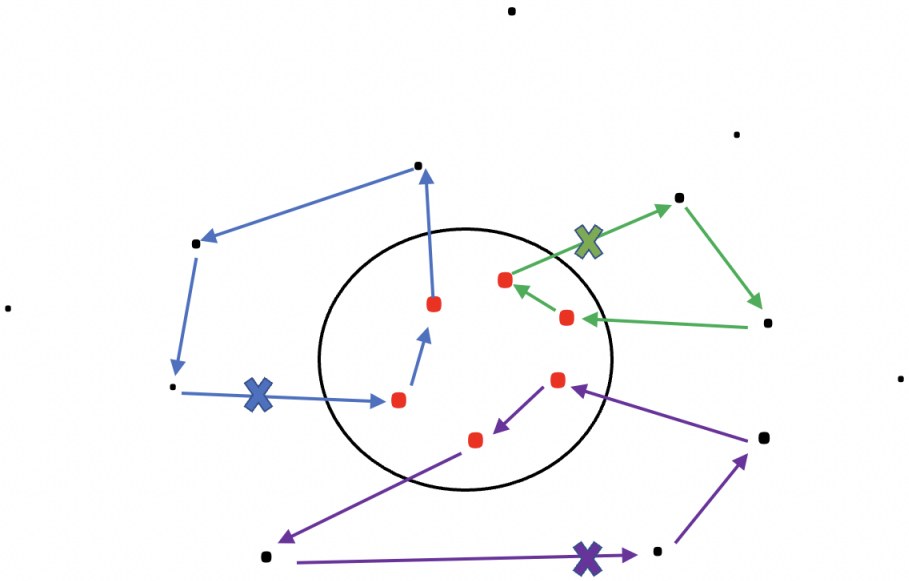


Figure 3.1: An instance of the HSCPP. The circular region is the HCN. Red and black nodes are inner and outer nodes, respectively. The size of a node indicates its prize. Directed arcs express the travel directions and the cross signs specify the starting points. Note that only a subset of nodes is visited.

represented as red dots in Figure 3.1. For the rest, we denote the set of nodes by V_{outer}, which are the black dots in Figure 3.1. In a patrol route, each node can be patrolled at most once. It is highly possible that each patrol route has a different travel length. To make all patrollers have a T-hours patrol, we introduce idle time, which is the difference between T hours and the travel time of the route. Because the HCN is more important, the idle time of a patroller is spent at one of the inner nodes in the route. The time a patroller spends in the HCN is computed by the sum of travel time between inner nodes and idle time. Note that the time spent in the HCN does not include the time traveling between the periphery of the HCN and the inner nodes. Therefore, our computation gives an underestimate of the actual time that a patroller spends in the HCN.

To distinguish among locations and choose the best ones to visit, we use prize P_i to quantify the importance of location i. Generally, the locations in the HCN are more important and should

have higher prizes than the ones not in the HCN. As shown in Figure 3.1, the red dots are larger than the black ones. The objective of the HSCPP is to maximize the sum of prizes collected among all patrol cars subject to the following constraints. The constraint related to the HCN is the constant presence requirement in the HCN. Other resource constraints include the number of patrollers (K) and the length of the patrol route (τ miles). We also assume that a patroller only enters and exits the HCN once, i.e., the patroller will cover the HCN during a continuous time window. Finally, we must decide on the starting points of the routes, which are the cross signs in Figure 3.1. Figure 3.1 shows a solution of the HSCPP by displaying three routes (blue, purple, and green). It specifies the locations to visit, the starting points, and the route directions. Before concluding this section, we have the following hardness result for the HSCPP.

Proposition 1 *The HSCPP is NP-hard.*

Given this NP-hardness, we develop several approaches for finding high-quality solutions for the HSCPP.

3.3 Solution Approaches of the HSCPP Routes

In this section, we develop solution approaches for the routing part of the HSCPP. In Section 3.3.1, we present a global approach and an IP formulation to find the routing solution for all patrol cars in a single run. In Section 3.3.2, we introduce the partition approach, where we partition the region geographically and solve the routing problem in each subregion separately using two IP formulations. In Section 3.3.3, we present a column generation approach.

3.3.1 Global Approach

The global approach of the HSCPP is an IP formulation. One of the challenges in the IP formulation is the absence of a police depot. In contrast to the HSCPP, a depot exists in the MCPRP and other variants of the VRP. More importantly, its existence is crucial for identifying the optimal tour and eliminating subtours. We circumvent the issue of no depot by observing that a patrol car must enter and exit the HCN exactly once for patrolling the HCN in a continuous time window. This leads to the requirement that the patrol car must visit at least one of the inner nodes. As a result, the set of inner nodes V_{inner} can function as a depot for each patrol car. For this purpose, we introduce a set D consisting of a copy of V_{inner}. We denote the nodes in D as virtual nodes. Each virtual node d_i corresponds to inner node $i \in V_{\text{inner}}$, such that $P_{d_i} = P_i$, $L_{ij} = L_{d_i j}$, and $L_{ji} = L_{j d_i}$ for any $j \in V$. With the help of D, an optimal tour can be identified easily because it contains exactly one of the nodes in D. We denote the union of V_{inner} and D by V^*_{inner}. We have the following binary decision variables. Let $x_{ijk} = 1$ if patrol car k visits arc (i, j); otherwise $x_{ijk} = 0$. Let $y_{ik} = 1$ if patrol car k visits node i; otherwise $y_{ik} = 0$. Our IP

formulation, denoted as IP3.1, is given below.

$$\text{IP3.1}: \quad \max \quad \sum_{i\in V\cup D}\sum_{k\in\mathcal{K}} P_i y_{ik} \tag{3.1}$$

$$\text{Subject to} \quad \sum_{j\in V\cup D} x_{jik} = y_{ik}, \quad \forall\, i \in V\cup D, k\in\mathcal{K}, \tag{3.2}$$

$$\sum_{k\in\mathcal{K}} y_{ik} \le 1, \quad \forall i \in V\cup D, \tag{3.3}$$

$$\sum_{j\in V\cup D} x_{ijk} - \sum_{j\in V\cup D} x_{jik} = 0, \quad \forall\, i \in V\cup D, k\in\mathcal{K}, \tag{3.4}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijk} \le \sum_{i\in S\setminus l} y_{ik}, \quad \forall S\subset V, l\in S, k\in\mathcal{K}, \tag{3.5}$$

$$\sum_{i\in V_{\text{outer}}}\sum_{j\in V^*_{\text{inner}}} x_{ijk} = 1, \quad \forall k\in\mathcal{K}, \tag{3.6}$$

$$\sum_{i\in V^*_{\text{inner}}}\sum_{j\in V_{\text{outer}}} x_{ijk} = 1, \quad \forall k\in\mathcal{K}, \tag{3.7}$$

$$\sum_{i\in D} y_{ik} = 1, \quad \forall k\in\mathcal{K}, \tag{3.8}$$

$$\sum_{k\in\mathcal{K}} y_{ik} + y_{d_i k} \le 1, \quad \forall\, i\in V_{\text{inner}}, \tag{3.9}$$

$$\sum_{i\in V\cup D}\sum_{j\in V\cup D} L_{ij}x_{ijk} \le \tau, \quad \forall\, k\in\mathcal{K}, \tag{3.10}$$

$$\begin{aligned}&(\textstyle\sum_{i\in V^*_{\text{inner}}}\sum_{j\in V^*_{\text{inner}}} L_{ij}x_{ijk} + \\ &(K\tau - \textstyle\sum_{i\in V\cup D}\sum_{j\in V\cup D}\sum_{k\in\mathcal{K}} L_{ij}x_{ijk}))/v \ge T,\end{aligned} \tag{3.11}$$

$$x_{ijk}\in\{0,1\}, \quad \forall\, i,j\in V\cup D, k\in\mathcal{K}, \tag{3.12}$$

$$y_{ik}\in\{0,1\}, \quad \forall\, i\in V\cup D, k\in\mathcal{K}. \tag{3.13}$$

In IP3.1, the objective function (3.1) maximizes the sum of prizes collected among all K patrol cars. Constraints (3.2) ensure that a location's prize is collected by patrol car k only after it is visited by patrol car k. Constraints (3.3) force each node to be visited by at most one patrol car. Constraints (3.4) are the flow-balance constraints at each node for all K tours. Constraints (3.5) enforce the subtour elimination constraints and are only applied to tours without a virtual node.

Constraints (3.6) and (3.7) together require that each patrol car enters and exits the HCN exactly once. Constraints (3.8) make sure that exactly one virtual node is visited by each patrol car. Note that this virtual node is picked to circumvent the lack of a given depot in the HSCPP. Constraints (3.9) ensure that given any pair of an inner node and its virtual correspondent, at most one of them can be visited in a tour. Constraints (3.10) limit the length of the tour for each patrol car to be less than τ. Constraint (3.11) requires that the sum of time spent in the HCN by all patrol cars is greater than the coverage duration in the HCN (T hours). Constraints (3.12) and (3.13) give the domains of our variables.

We have an exponential number of subtour elimination constraints in (3.5). We only add these constraints as needed, i.e., whenever there is a subtour that does not include a virtual node. When our decision variables x_{ijk} and y_{ik} take integer values, subtours can be identified easily by enumerating the cycles induced by decision variables x_{ijk} with values of one. However, when Gurobi solves the linear programming (LP) relaxation of IP3.1 using the branch-and-bound algorithm, x_{ijk} and y_{ik} can take fractional values. In this case, we denote the LP relaxation solution of IP3.1 by (x^*_{ijk}, y^*_{ik}) for some patrol car $k \in \mathcal{K}$. Then, constraints (3.5) are violated if there exists a subset $S' \subset V$ and a node $l' \in S'$ such that $\sum_{i \in S'} \sum_{j \in S'} x^*_{ijk} > \sum_{i \in S' \setminus l'} y^*_{ik}$. We find such a subset S' and node l' by solving an integer program. Let $z_i = 1$ if node $i \in S'$; otherwise $z_i = 0$. Similarly, let $w_{ij} = 1$ if $i \in S'$ and $j \in S'$; otherwise $w_{ij} = 0$. Moreover, let

$v_i = 1$ if $i = l'$; otherwise $v_i = 0$. Our separation procedure, denoted by SIP3.1, is given below.

$$\text{SIP3.1}: \quad \max \quad \sum_{i\in V}\sum_{j\in V} x^*_{ijk} w_{ij} - \sum_{i\in V} y^*_{ik} z_i + \sum_{i\in V} y^*_{ik} v_i \tag{3.14}$$

$$\text{subject to} \quad w_{ij} \le z_i, \quad \forall i \in V, j \in V, \tag{3.15}$$

$$w_{ij} \le z_j, \quad \forall i \in V, j \in V, \tag{3.16}$$

$$v_i \le z_i, \quad \forall i \in V, \tag{3.17}$$

$$\sum_{i\in V} v_i = 1, \tag{3.18}$$

$$w_{ij} \in \{0,1\}, \quad \forall i \in V, j \in V, \tag{3.19}$$

$$z_i, v_i \in \{0,1\}, \quad \forall i \in V. \tag{3.20}$$

In SIP3.1, the objective function (3.14) maximizes the difference between $\sum_{i\in S'}\sum_{j\in S'} x_{ijk}$ and $\sum_{i\in S'\setminus l'} y_{ik}$. Constraints (3.15) and (3.16) ensure that arc (i, j) is included if and only if nodes i and j are included in S'. Constraints (3.17) and (3.18) require that only one node in S' is selected as node l'. Constraints (3.19) and (3.20) give the domain of our variables.

If the resulting objective of the SIP3.1 is greater than zero, we add the constraints $\sum_{i\in S'}\sum_{j\in S'} x_{ijk} \le \sum_{i\in S'\setminus l'} y_{ik}$ into IP3.1. After SIP3.1 is applied to each patrol car, we solve the LP relaxation of IP3.1 again. The process of adding subtour elimination constraints and solving the LP relaxation of IP3.1 is repeated until the objective of SIP3.1 is less than or equal to zero for all K patrol cars. In our implementation, SIP3.1 is only applied to the root node in the branch-and-bound process.

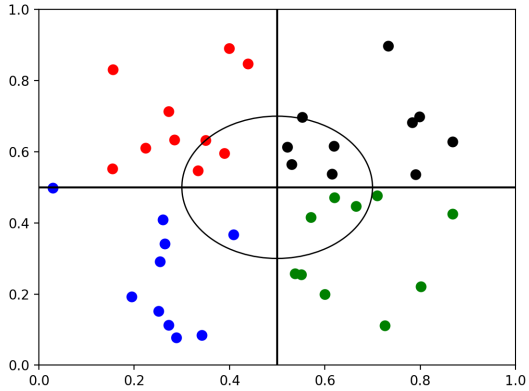


Figure 3.2: An example illustrating the partition of the region of interest (unit square). The circular region is the HCN. The colors of the nodes differentiate the subregions.

3.3.2 Partition Approach

Although IP3.1 can solve the routing part of the HSCPP in a single run, the optimality gap, as we will see in Section 3.5, has a large variance. For some instance, the optimality gap is well over 20%, after an hour of runtime. To address this issue, we develop the partition approach. We partition our region of interest geographically into K subregions. Each patrol car is assigned to one of the subregions and must only visit nodes in the subregion. Such partitioning offers several advantages. First, the patrol task of an area is often divided geographically [38]. Second, patrollers often prefer to travel in a compact area of the region that they are more familiar with. This way, when an emergency call goes out, the patrollers can respond to the case faster [11]. Third, partitioning makes the problem size of each subproblem notably smaller and leads to better computational efficiency.

We discuss our method of partitioning here. Because the HCN is a circular region, we may construct K partitions of equal size and shape. For the default partition, we set the center of the HCN as the origin and fix one of the boundaries between subregions as the negative y-axis.

Then, the other boundaries are also known because the angle between adjacent boundaries is $2\pi/K$. Lastly, we extend the boundaries to the edges of our region of interest. Figure 3.2 shows the partitioning for $K = 4$ when the region of interest is the unit square. The HCN is the circle centered at (0.5, 0.5) with a radius of 0.2. The nodes are potential patrol locations.

After partitioning the region, we build the solution of the HSCPP by applying IP3.1 to each subregion. The main issue lies with constraint (3.11). We can naively add the constraint that the time spent in the HCN for each patrol car is greater than T/K. However, this causes a patrol car in a subregion with fewer inner nodes to have a longer idle time and consequently collect fewer prizes. To avoid this problem, we change the objective function to minimize the length of the tour and add a constraint that the patrol car must collect at least $\theta\%$ of the total prizes in the subregion. When θ is small, because the inner nodes have higher prizes and the tour must enter and exit the HCN once, the optimal tour will be concentrated around the HCN. The length of the optimal tour is short and the majority of patrol time in the HCN is spent idling. When the majority of inner nodes in the subregion are visited and one still wants to increase θ, the optimal tour will be forced to include more outer nodes. The inclusion of more outer nodes increases the length of the tour and decreases the time spent in the HCN because the idle time decreases. As a result, we can use θ to control the length of the tour and the time spent in the HCN. More importantly, θ can be different among the subregions. The patrol car in the subregion with more inner nodes can stay in the HCN longer, whereas the patrol car in the subregion with fewer inner nodes can visit more outer nodes.

Before giving the IP formulation, we introduce some notation. We let V_k be the set of nodes in subregion k. D_k is the set of virtual nodes in subregion k. $V_{\text{inner},k}$ and $V_{\text{outer},k}$ are the set of inner and outer nodes, respectively, in subregion k. $V^*_{\text{inner},k}$ denotes the union of $V_{\text{inner},k}$ and D_k.

We have the following binary decision variables. Let $x_{ij} = 1$ if the patrol car visits arc (i, j); otherwise $x_{ij} = 0$. Let $y_i = 1$ if the patrol car visits node i; otherwise $y_i = 0$. Our IP formulation for subregion k, denoted as IP3.2, is given below.

$$\text{IP3.2}: \quad \min \quad \sum_{i \in V_k \cup D_k} \sum_{j \in V_k \cup D_k} L_{ij} x_{ij} \tag{3.21}$$

$$\text{subject to} \quad \sum_{j \in V_k \cup D_k} x_{ji} = y_i, \quad \forall\, i \in V_k \cup D_k, \tag{3.22}$$

$$\sum_{j \in V_k \cup D_k} x_{ij} - \sum_{j \in V_k \cup D_k} x_{ji} = 0, \quad \forall\, i \in V_k \cup D_k, \tag{3.23}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq \sum_{i \in S \setminus l} y_i, \quad \forall S \subset V_k, l \in S, \tag{3.24}$$

$$\sum_{i \in V_{\text{outer},k}} \sum_{j \in V^*_{\text{inner},k}} x_{ij} = 1, \tag{3.25}$$

$$\sum_{i \in V^*_{\text{inner},k}} \sum_{j \in V_{\text{outer},k}} x_{ij} = 1, \tag{3.26}$$

$$\sum_{i \in D_k} y_i = 1, \tag{3.27}$$

$$y_i + y_{d_i} \leq 1, \quad \forall\, i \in V_{\text{inner},k}, \tag{3.28}$$

$$\sum_{i \in V_k \cup D_k} P_i y_i \geq \tfrac{\theta}{100} \sum_{i \in V_k} P_i, \tag{3.29}$$

$$x_{ij} \in \{0, 1\}, \quad \forall\, i, j \in V_k \cup D_k, \tag{3.30}$$

$$y_i \in \{0, 1\}, \quad \forall\, i \in V_k \cup D_k, \tag{3.31}$$

In IP3.2, the objective function (3.21) minimizes the length of the tour. Constraint (3.29) ensures that the patrol car collects at least $\theta\%$ of total prizes in the subregion. The remaining constraints in IP3.2 are similar to those in IP3.1. We run IP3.2 for $\theta \in \{70 + 3l : l = 0, 1, 2, \ldots, 10\}$ in each subregion. If the objective of IP3.2 is greater than the distance limit τ for some θ, we discard the IP3.2 tour for that θ.

To pick a θ for each subregion and satisfy the constant presence requirement in the HCN,

we introduce a simple IP formulation. Let P_{kl} and IT_{kl} denote the prizes collected and time spent in the HCN of the IP3.2 tour in the kth subregion using the lth θ, where $l = \{0, 1, 2, \ldots, 10\}$ correspond with $\theta = \{70, 73, 76, \ldots, 100\}$, respectively. Note that $P_{kl} = IT_{kl} = 0$ if the IP3.2 tour in the kth subregion using the lth θ is discarded. Let $X_{kl} = 1$ if we pick the lth θ for the kth subregion; otherwise $X_{kl} = 0$. Our IP formulation of picking θ for each subregion, denoted as IP3.3, is given below.

$$\text{IP3.3}: \quad \max \quad \sum_{k=1}^{K} \sum_{l=0}^{10} P_{kl} X_{kl} \tag{3.32}$$

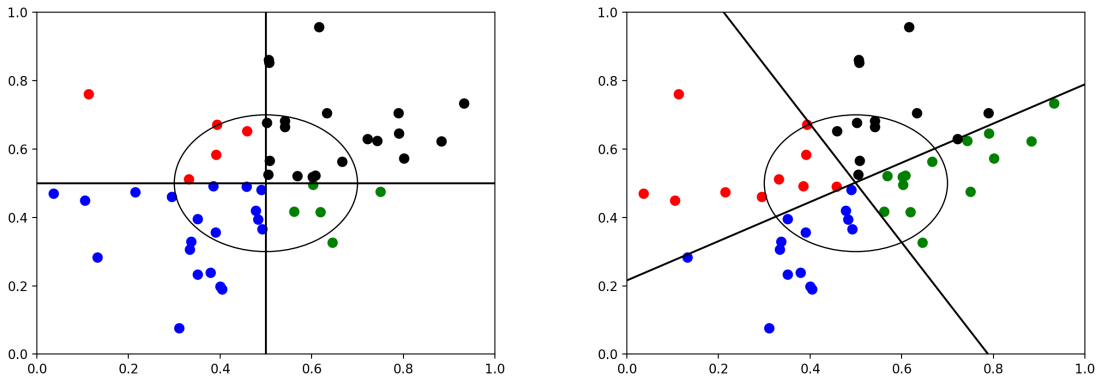
$$\text{subject to} \quad \sum_{l=0}^{10} X_{kl} = 1, \quad \forall k \in \{1, 2, \ldots, K\} \tag{3.33}$$

$$\sum_{k=1}^{K} \sum_{l=0}^{10} IT_{kl} X_{kl} \geq T, \tag{3.34}$$

$$X_{kl} \in \{0, 1\}, \quad \forall k \in \{1, 2, \ldots, K\}, l \in \{0, 1, 2, \ldots, 10\}. \tag{3.35}$$

In IP3.3, the objective function (3.32) maximizes the sum of prizes collected among all patrol cars. Constraints (3.33) force that exactly one θ is picked for a subregion. Constraint (3.34) ensures that the time spent in the HCN among all subregions is greater than the coverage duration in the HCN. Constraints (3.35) give the domain of variables.

So far, when partitioning the region, we have followed the rule that one of the boundaries must be the negative y-axis. This partition could work well when the number of nodes is similar between subregions. However, when there is an imbalance in the number of nodes (see Figure 3.3a), this partitioning would result in sub-optimal routing solutions. Specifically, the region with more nodes would have many uncovered nodes due to the distance constraint. One way to correct this is to introduce a rotation among the boundaries of the subregions. We rotate the boundaries by an angle of σ in the counterclockwise direction. Note that the range of σ is between 0 and



(a) (b)

Figure 3.3: An instance illustrating the imbalance of the default partition. Panel (a) shows the imbalance of node distributions between subregions if the default partition is applied. Panel (b) shows a more balanced partition if the boundaries are rotated by $\pi/6$.

$2\pi/K$ because we are back to the original case ($\sigma = 0$) when the rotation angle is $2\pi/K$. In Figure 3.3b, we plot the rotated subregions when $\sigma = \pi/6$. After rotating, a new partition is formed and IP3.2 and IP3.3 can be reapplied to find the new routing solution. We denote the set of solutions generated by all partitions as partition solutions. Among partition solutions, the solution with the largest sum of prizes collected is called the best partition solution (BPS).

3.3.3 Column Generation (CG) Approach

The solution quality of the partition approach mainly depends on the number of rotations and the set of θ. It may be time-consuming to find the optimal partitioning. More importantly, using the partition approach eliminates the possibility of two patrol cars sharing the patrolling of a subregion with many hot spots. One method to improve the partition approach is to apply column generation using the partition solutions as the initial columns.

Let $\mathcal{R}$ be the set of all possible patrol routes. In particular, the routes in $\mathcal{R}$ have travel distances less than τ and enter and exit the HCN exactly once. For each $r \in \mathcal{R}$, let $a_{ir} = 1$ if

route r visits node i and $a_{ir} = 0$ otherwise. Let $z_r = 1$ if route r is used in the CG solution; otherwise $z_r = 0$. The amount of prizes collected by route r is denoted by $c_r = \sum_{i \in V} a_{ir} P_i$. IT_r gives the time spent in the HCN of route r. The master problem of the CG has the following IP formulation.

$$\text{IP3.4}: \quad \max \quad \sum_{r \in \mathcal{R}} c_r z_r \tag{3.36}$$

$$\text{subject to} \quad \sum_{r \in \mathcal{R}} a_{ir} z_r \leq 1, \quad \forall\, i \in V, \tag{3.37}$$

$$\sum_{r \in \mathcal{R}} IT_r z_r \geq T, \tag{3.38}$$

$$\sum_{r \in \mathcal{R}} z_r = K, \tag{3.39}$$

$$z_r \in \{0, 1\}, \quad \forall\, r \in \mathcal{R}. \tag{3.40}$$

The objective function (3.36) maximizes the sum of prizes collected among all patrol cars. Constraints (3.37) ensure that each node is visited at most once. Constraint (3.38) forces the sum of time spent in the HCN to be greater than the coverage duration in the HCN. Constraint (3.39) makes sure that K tours are chosen in the CG solution because we have K patrol cars. Constraints (3.40) give the domain of our decision variables.

Let IP3.4R be the restricted version of IP3.4 such that IP3.4R is identical to IP3.4 except that it only contains a subset of the z variables. In particular, we denote the subset of the z

variables by $\mathcal{R}'$. We have

$$\text{IP3.4R}: \quad \max \quad \sum_{r\in\mathcal{R}'} c_r z_r \tag{3.41}$$

$$\text{subject to} \quad \sum_{r\in\mathcal{R}'} a_{ir} z_r \leq 1, \quad \forall\, i \in V, \tag{3.42}$$

$$\sum_{r\in\mathcal{R}'} IT_r z_r \geq T, \tag{3.43}$$

$$\sum_{r\in\mathcal{R}'} z_r = K, \tag{3.44}$$

$$z_r \in \{0,1\}, \quad \forall\, r \in \mathcal{R}'. \tag{3.45}$$

By relaxing the integrality constraint (3.45), we denote the LP relaxation of IP3.4R by LP3.4R.

It is impractical to write out all routes in $\mathcal{R}$ explicitly because there is an exponential number of them. To work around this issue, we dynamically add tours to LP3.4R via a pricing procedure. Let $\boldsymbol{\alpha}$, β, and γ be the dual variables associated with constraints (3.42), (3.43), and (3.44). Then, we have ($\boldsymbol{\alpha}$*, β^*, γ^*) as a dual optimal solution of corresponding variables when solving LP3.4R. When adding tours to LP3.4R, we especially want to add tours with a positive reduced cost because only these tours can increase the objective value of LP3.4R. The reduced cost RC_r of tour r can be expressed by

$$RC_r = c_r - \sum_{i\in V} a_{ir}\alpha_i^* + IT_r\beta^* - \gamma^*. \tag{3.46}$$

In the pricing problem, we find a tour with a positive reduced cost by maximizing RC_r with an IP formulation. We have the following binary decision variables. Let $x_{ij} = 1$ if the patrol car visits arc (i, j); otherwise $x_{ij} = 0$. Let $y_i = 1$ if the patrol car visits node i; otherwise $y_i = 0$. Recall that D is the set of virtual nodes corresponding to a copy of V_{inner}. Our IP formulation for

the pricing problem, denoted as IP3.5, is given below.

$$\text{IP3.5}: \quad \max \quad \sum_{i\in V\cup D} P_i y_i - \sum_{i\in V\cup D} \alpha_i^* y_i$$
$$+\beta^*\Big(\sum_{i\in V^*_{\text{inner}}}\sum_{j\in V^*_{\text{inner}}} L_{ij}x_{ij} + \tau - \sum_{i\in V\cup D}\sum_{j\in V\cup D} L_{ij}x_{ij}\Big)/v$$
$$-\gamma^*. \tag{3.47}$$

$$\text{subject to} \quad \sum_{j\in V\cup D} x_{ji} = y_i, \quad \forall\, i \in V\cup D, \tag{3.48}$$

$$\sum_{j\in V\cup D} x_{ij} - \sum_{j\in V\cup D} x_{ji} = 0, \quad \forall\, i \in V\cup D, \tag{3.49}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \le \sum_{i\in S\setminus l} y_i, \quad \forall S \subset V, l \in S, \tag{3.50}$$

$$\sum_{i\in V_{\text{outer}}}\sum_{j\in V^*_{\text{inner}}} x_{ij} = 1, \tag{3.51}$$

$$\sum_{i\in V^*_{\text{inner}}}\sum_{j\in V_{\text{outer}}} x_{ij} = 1, \tag{3.52}$$

$$\sum_{i\in V\cup D}\sum_{j\in V\cup D} L_{ij}x_{ij} \le \tau, \tag{3.53}$$

$$\sum_{i\in D} y_i = 1, \tag{3.54}$$

$$y_i + y_{d_i} \le 1, \quad \forall\, i \in V_{\text{inner}}, \tag{3.55}$$

$$x_{ij} \in \{0,1\}, \quad \forall\, i,j \in V\cup D, \tag{3.56}$$

$$y_i \in \{0,1\}, \quad \forall\, i \in V\cup D. \tag{3.57}$$

Objective (3.47) maximizes the reduced cost. The constraints in IP3.5 are similar to the constraints in IP3.1 and IP3.2. If the optimal objective of IP3.5 is positive, we add a variable z reflecting the optimal solution of the pricing problem. Then, LP3.4R is solved to update the dual solutions. At this point, IP3.5 is solved again. We repeatedly add z variables until the objective of IP3.5 is non-positive. When that happens, we run IP3.4R with the updated $\mathcal{R}'$ to find the CG solution.

3.3.3.1 Enhancements of Column Generation

The pricing problem of the CG involves solving an IP problem with $|V| + |D|$ nodes. Finding the exact solution of such a problem can be computationally expensive when $|V|+|D|$ is large. In addition, we need to solve the pricing problem for each iteration of the CG. Considering that it could take many iterations for the CG to terminate, we conclude that the current pricing model may not be ideal. As we see in the computational experiment section, the CG solution will not improve from the partition solutions if IP3.5 is applied.

Elementary Shortest Path For Pricing: The pricing procedure of a vehicle routing problem can often be modeled as an elementary shortest path problem with resource constraints (ESPPRC). The only issue here is to determine the source and sink of the ESPPRC. In the VRP, the source and the sink are the depot. In the HSCPP, we recall that the set of inner nodes functions as the depot. In other words, there are $|V_{\text{inner}}|$ choices of the source and the sink. Assume that we pick inner node i^*. Then, the source s is an artificial node with only outgoing arcs, which are identical to node i^*'s outgoing arcs. Similarly, the sink c is another artificial node with only incoming arcs, which are identical to node i^*'s incoming arcs. We update the nodes in the network by setting $V_{i^*} = V \setminus \{i^*\} \cup \{s, t\}$ and $V_{\text{inner},i^*} = V_{\text{inner}} \setminus \{i^*\} \cup \{s, t\}$. The weight w_{ij} of arc (i, j) is defined as follows:

$$w_{ij} = \begin{cases} \alpha_i^* - P_i, & \text{for } i, j \in V_{\text{inner},i^*}, \\ \alpha_i^* - P_i + \beta^* L_{ij}/v, & \text{otherwise.} \end{cases} \tag{3.58}$$

Note that w_{ij} describes the negation of (3.47), omitting constants. There are two resource constraints in the ESPPRC. First, the length of the shortest path should be shorter than τ. Second, the patrol car can only exit the HCN once. Because the shortest path starts and ends at inner

node i^*, forcing the patrol car to exit the HCN once in the ESPPRC is equivalent to forcing the patrol car to enter and exit the HCN once in the HSCPP. A detailed IP formulation is included in Appendix B. After finding the optimal shortest path S, we find the reduced cost of S by summing $-\sum_{(i,j)\in S} w_{ij}x_{ij}$ and $T\beta^* - \gamma^*$.

To solve the pricing problem optimally, we need to solve the ESPPRC for every inner node in a CG iteration. However, rather than solving them optimally, it suffices to find a tour with a positive reduced cost and stop solving the ESPPRC for the rest of the inner nodes. As we will see in our computational experiments, this massively reduces the runtimes of the CG. Modeling the pricing problem as the ESPPRC has other advantages. First, the ESPPRC is a well-studied problem and methods such as the bidirectional labeling algorithm [59] and the pulse algorithm [42] can be used to solve the ESPPRC efficiently. Second, these methods can be much faster than solving the ESPPRC using Gurobi.

Dual-Optimal Inequalities: One issue of the CG is that there often exists a large oscillation in the values of the dual variables between CG iterations [5]. This slows down the convergence of the CG. To tackle this issue, we add the following dual-optimal inequalities (DOIs) to stabilize the CG.

Proposition 2 *Given a pair of nodes $i, j \in V$ that satisfies $a_{ir} = a_{jr},\ \forall r \in \mathcal{R}'$, then there exists a dual-optimal solution satisfying $\alpha_i^* = \alpha_j^*$.*

We implement Proposition 2 in the following way. For a pair of nodes (i, j) that satisfies $a_{ir} = a_{jr}, \forall r \in \mathcal{R}'$, we add $\alpha_i = \alpha_j$ to the dual of LP3.4R. Then, we use the resulting dual-optimal solution in the pricing procedure of the CG to improve computational efficiency.

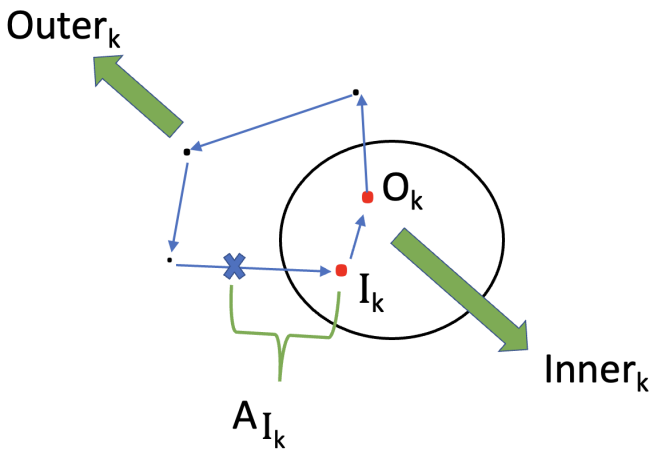


Figure 3.4: An example illustrating I_k, O_k, A_{I_k}, Inner$_k$, and Outer$_k$. The circular region is the HCN. Red and black nodes are inner and outer nodes, respectively. Directed arcs indicate the directions of travel and the cross sign specifies the starting points. The travel time in the tour from the cross sign to I_k is A_{I_k}.

3.4 Finding the Starting Points of the HSCPP Routes

In this section, we find the starting points of the tours found in Section 3.3 and ensure that there exists at least one patrol car in the HCN at any given time during the patrol.

Given a tour k, we denote node O_k as the inner node that has an outgoing arc to an outer node. Similarly, node I_k is defined as the inner node that has an incoming arc from an outer node. Because a patrol car enters and exits the HCN once, nodes O_k and I_k are unique. Note that O_k and I_k can be identical if there is only one inner node in tour k. Using I_k and O_k, we break tour k into two parts. The inner part of tour k denoted by Inner$_k$ is defined as the part from I_k to O_k. Similarly, the outer part of tour k denoted by Outer$_k$ is defined as the part from O_k to I_k. Figure 3.4 shows an example of O_k, I_k, Inner$_k$, and Outer$_k$. We have the following claim.

Claim 1 *For $1 \leq k \leq K$, the starting point of tour k is in Outer$_k$.*

Claim 1 ensures that a patrol car covers the HCN in a continuous time window. Before

presenting our method of finding the starting points, we introduce some notation. We denote the set of nodes in tour k by $\mathcal{N}_k$. Without loss of generality, we assume that patrol car k visits $\mathcal{N}_k$, with $1 \leq k \leq K$. We let A_{I_k} and D_{O_k} be the arrival time at node I_k and the departure time from node O_k, respectively. In other words, A_{I_k} and D_{O_k} are the arrival and departure times in the HCN of patrol car k, respectively. Because the time spent in the HCN is fixed, we have the relationship that $D_{O_k} = A_{I_k} + IT_k$. Moreover, we denote t_{j,I_k} as the travel time in tour k from node $j \in \mathcal{N}_k$ to I_k.

Now, we present the method to find the starting point. It is important to note that A_{I_k} has a one-to-one correspondence with the starting point. Given A_{I_k}, we find the starting point in the following way. We start from I_k and trace backwards through tour k so that the travel time from the starting point to I_k is A_{I_k} (see Figure 3.4). The starting point is on arc (i, j) if $t_{i,I_k} \leq A_{I_k} \leq t_{j,I_k}$. The particular location on arc (i, j) can be computed easily because the patrol car travels at a uniform speed. As a result, the problem of finding the starting point is reduced to determining A_{I_k}.

To determine A_{I_k}, we rely on the constant presence requirement in the HCN. A sufficient condition for the requirement is that patrol car $k + 1$ enters the HCN before patrol car k departs the HCN for $1 \leq k \leq K - 1$. This leads to $A_{I_{k+1}} \leq D_{O_k}$. To force the starting point in Outer$_{k+1}$, we have $A_{I_{k+1}} \leq (T - IT_{k+1})$. Combining these two inequalities, we set $A_{I_{k+1}} = \min(D_{O_k}, T - IT_{k+1})$. The starting points of patrol cars 1 and K are fixed because patrol car 1 covers the HCN at the beginning of the patrol (0th hour) whereas patrol car K covers the HCN at the end of the patrol (Tth hour). Hence, patrol car 1 starts at I_1, visits inner nodes immediately, and exits the HCN for outer nodes. Patrol car K starts at O_K, moves away from the HCN, and visits inner nodes at the end of its tour. Because patrol car K is leaving the HCN from O_K, the

Patrol Car	A_{I_k}	D_{I_k}
1	0	IT_1
2	$\min(IT_1, T - IT_2)$	$\min(T, IT_1 + IT_2)$
3	$\min(IT_1 + IT_2, T - IT_3)$	T

Table 3.1: An example illustrating the arrival and departure times of the patrol cars in the HCN, such that a patrol car is always in the HCN during the patrol.

coverage of the HCN will not start until it visits I_K. For patrol cars $2 \leq k \leq K-1$, the starting points are determined in ascending order using the rule that $A_{I_{k+1}} = \min(D_{O_k}, T - IT_{k+1})$. Table 3.1 shows an example of the arrival and departure times in the HCN when there are $K = 3$ patrol cars.

3.5 Case Study: Montgomery County, Maryland

To test and compare the performance of our solution approaches, we perform a series of computational studies based on real-world data. The main goal of the computational experiment is to compare the solution qualities of the proposed approaches: the global approach (IP3.1), the partition approach (IP3.2 + IP3.3), and the CG approach. Additionally, we consider the fourth approach: the global approach with a warm start, which uses the best partition solution as an initial solution for IP3.1. We make these comparisons on a set of 25 simulated instances based on real crime data found in dataMontgomery (`https://data.montgomerycountymd.gov/Public-Safety/Crime/icn6-v9z3`). Note that dataMontgomery is the county's statistical repository. All computations are performed on a computer with an Apple M1 processor and 8 GB of RAM. The IP models are solved by Gurobi 9.1.2 via Python API.

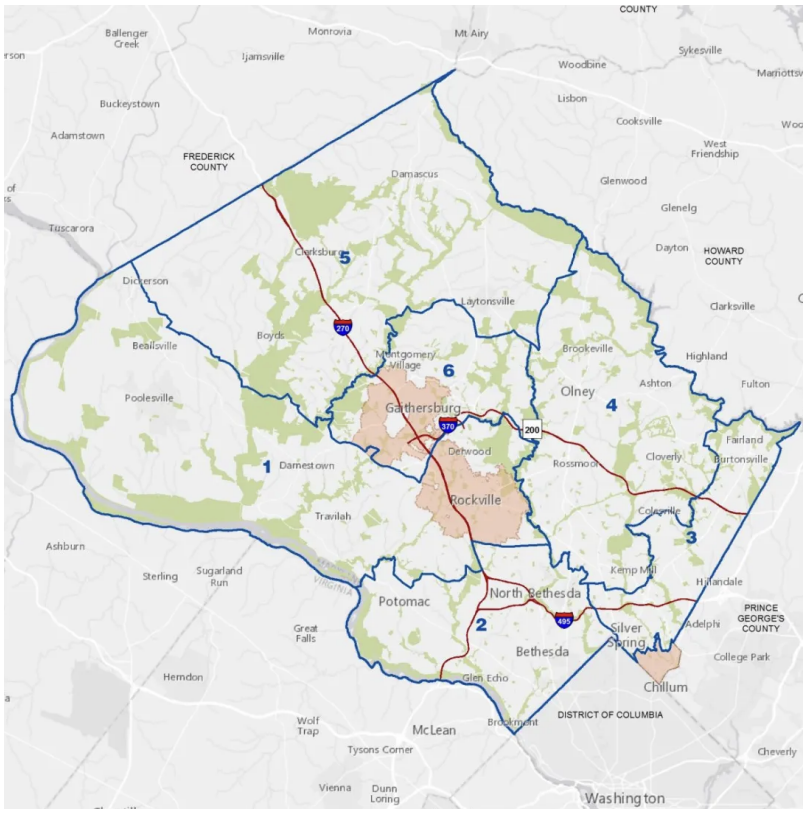


Figure 3.5: District map of Montgomery County. The districts are separated by solid blue lines. Image retrieved as screen capture from `https://mococivilrights.wordpress.com/montgomery-county/maps/`.

3.5.1 Data and Instance Description

Montgomery County is a county in the state of Maryland, with a population of around one million. It consists of six districts (see Figure 3.5). Based on dataMontgomery, 2809 second-degree assaults happened in Montgomery County in 2021. Table 3.2 shows the number of assaults in each district. We use the 6th district in our computational experiments because it was tied for the most second-degree assaults and consists of urban areas such as Gaithersburg and Montgomery Village.

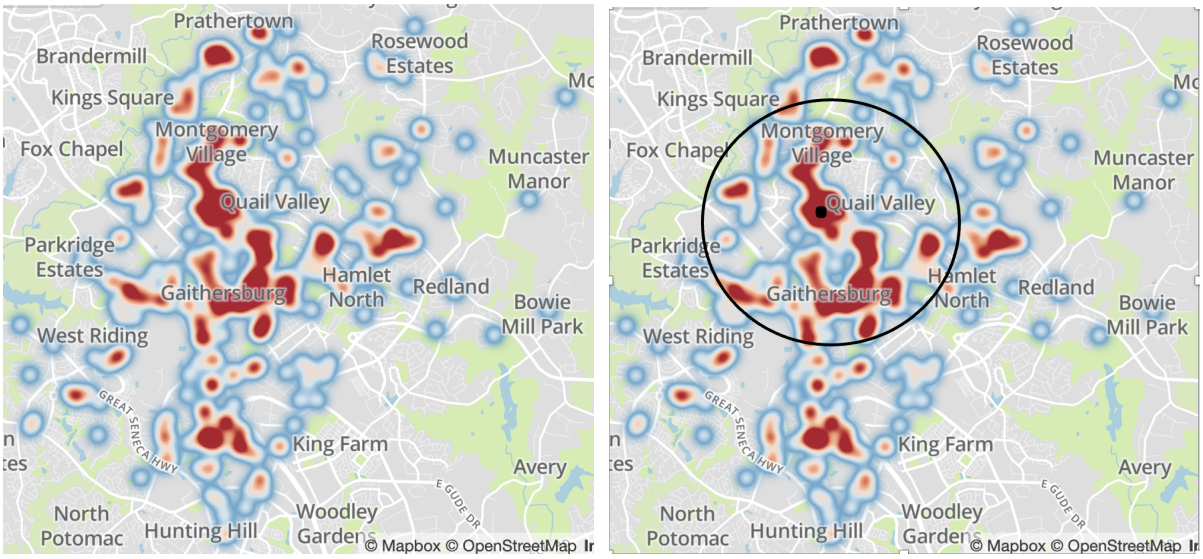
In Figure 3.6a, we plot the heat map of second-degree assaults in the 6th district during 2021. In this heat map, darker colors indicate more crime. We observe that second-degree as-

District	No. of second-degree assaults
1	295
2	271
3	627
4	485
5	504
6	627
Total	2809

Table 3.2: Number of second degree assaults committed in Montgomery County during 2021.

saults are concentrated in Gaithersburg and Montgomery Village, so we use these two locations as the HCN. In particular, our HCN is represented by the circle centered at (39.16021^o, -77.20217^o) latitude and longitude. Based on the heat map and the data, we set the radius of the HCN to 1.4 miles. Figure 3.6b shows the resulting HCN. Observe that the center of the HCN is near the middle point between Gaithersburg and Montgomery Village, such that the HCN includes both locations.

We consider two ways to generate patrol locations. On the one hand, we can use the center (39.16021^o, -77.20217^o) of the HCN and generate patrol locations uniformly within a 5-mile radius. However, this will produce many nodes outside the region of interest. On the other hand, we can generate patrol locations using four smaller circles. These circles are centered at (39.16021^o, -77.20217^o), (39.18777^o, -77.18677^o), (39.15905^o, -77.16044^o), and (39.12276^o, -77.21557^o) and include shopping malls, grocery stores, and restaurants where crimes are more likely to happen. Each circle has a radius of 1.4 miles (following the way we define the area of the HCN) and we generate 20 locations uniformly in each circle. Now, only one of the circles can include locations that fall outside the boundaries of the region (see Figure 3.7). If such a



(a) (b)

Figure 3.6: Panel (a) shows the heat map of second-degree assaults in the 6th district during 2021. Panel (b) gives our circular HCN including Gaithersburg and Montgomery Village.

location is generated, we disregard it and generate a new one. In our experiment, we use the second approach to generate patrol locations. In total, there are 80 patrol locations. We use 80 locations to model the situation where a patroller visits 4-6 nodes per hour. When $K = 3$, 60%-90% locations can be visited in the patrol. In Figure 3.7, we plot a distribution of the patrol locations.

After generating the nodes, we compute the network distance between each pair of nodes using Google Map Application Programming Interface (API). When computing the distance, we prevent the patroller from using highways. Thus, in our computational experiments, L_{ij} is the real-world travel distance computed by Google Map API. For each set of nodes, we generate five instances using a prize distribution, i.e., these five instances share the same distance matrix but the prizes associated with their nodes differ. The prize distribution is the uniform distribution between 0 and 1. In addition, if $i \in V_{\text{inner}}$, $P_i = 1$. Overall, in the computational experiments, we have five sets of nodes and 25 instances in total.

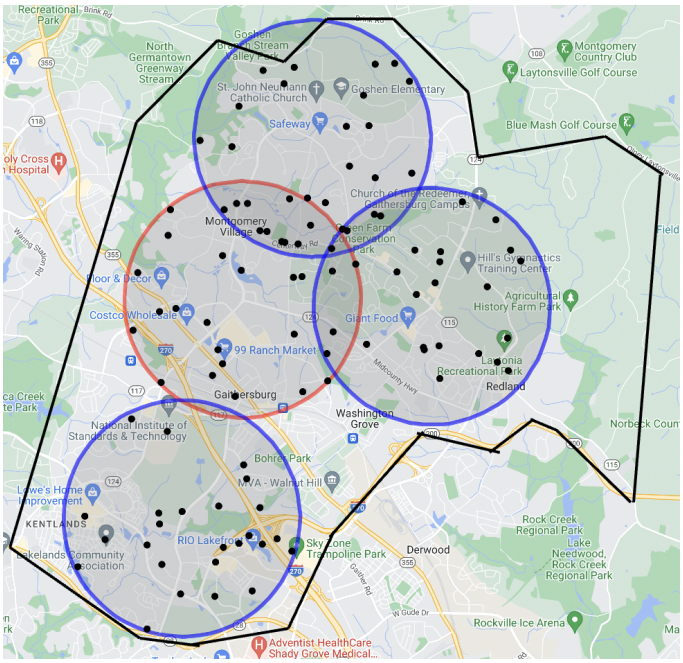


Figure 3.7: A distribution of patrol locations (black dots). The red circle is the HCN and the black solid lines indicate the boundaries of the district.

Based on an interview with a Lake Oswego police officer [37], a patrol car can idle for 3-5 hours during a 10-hour shift and the average patrol speed is 5-10 miles per hour. Therefore, we set the patrol speed v to be 5.2 miles per hour because urban area patrol usually involves significant idling. In addition, we set the time limit T and distance limit τ of a patrol route to be 4 hours and 20.8 miles, respectively. It is common that patrollers need to visit a location multiple times in a shift. For a 4-hour route in an 8-hour shift, patrollers have the option to tour the same route twice. Finally, we set the number of patrol cars K to be 3.

3.5.2 Computational Performance of the Proposed Approaches

We compare the computational performance and solution qualities between the proposed approaches: the global approach (IP3.1), the partition approach (IP3.2 + IP3.3), and the CG approach. Additionally, we consider the fourth approach: the global approach with a warm start,

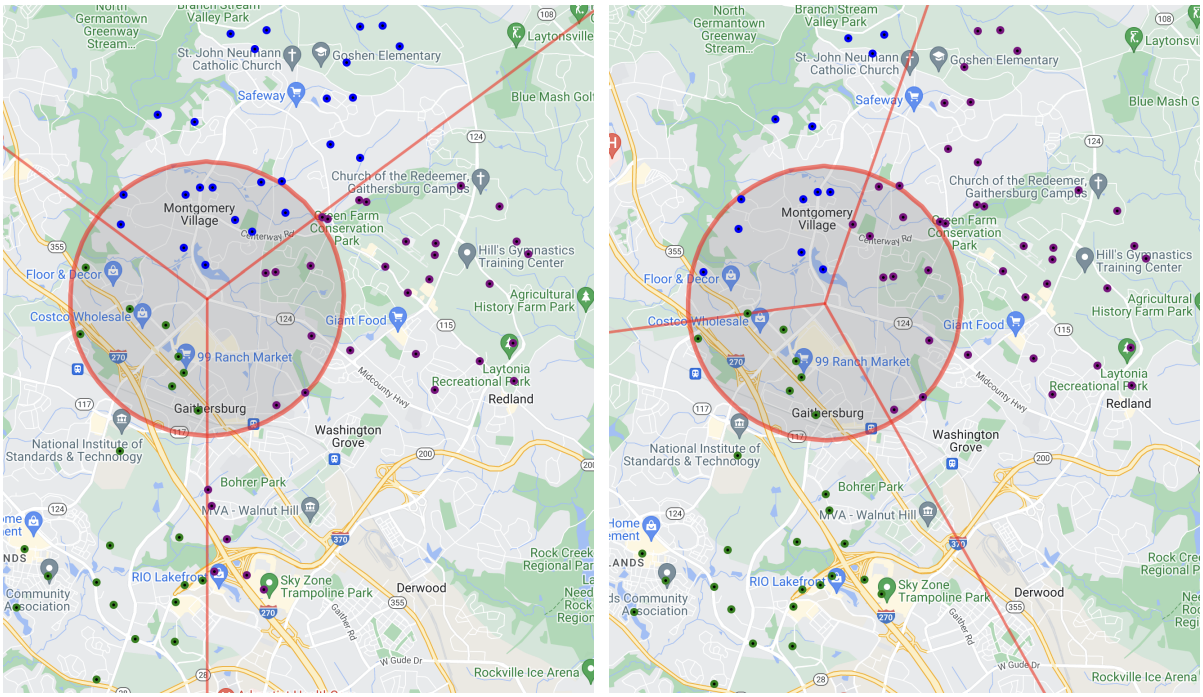
	Global Approach (1-hour runtime)			Partition Approach			CG Approach			Global w. Warm Start (3000s runtime)		
ID	Prize	UB	Gap	Prize	Gap	RT(s)	Prize	Gap	RT(s)	Prize	UB	Gap
1	48.21	49.10	1.81%	47.89	2.46%	133.4	47.89	2.46%	2,212.4	48.29	49.07	1.59%
2	48.80	50.66	4.47%	49.61	2.07%	126.1	50.09	1.13%	2,213.2	49.84	50.95	2.18%
3	49.92	51.68	3.41%	50.47	2.34%	105.1	50.47	2.34%	3,000.0	51.33	51.66	0.64%
4	49.47	50.54	2.12%	48.70	3.64%	138.6	49.67	1.72%	2,898.2	50.24	50.56	0.63%
5	50.99	51.87	1.69%	50.37	2.89%	137.9	50.41	2.81%	2,378.3	51.39	51.87	0.93%
6	48.57	49.21	1.31%	46.89	4.71%	188.3	48.54	1.36%	1,702.0	47.94	49.24	2.64%
7	48.91	49.40	0.99%	47.71	3.42%	301.6	48.52	1.78%	1,936.2	48.95	49.37	0.85%
8	44.12	49.25	10.41%	47.23	4.10%	188.2	47.95	2.64%	1,939.2	47.99	49.17	2.40%
9	45.62	47.03	3.00%	44.26	5.89%	254.2	46.57	0.98%	2,108.7	46.22	47.09	1.85%
10	46.71	47.97	2.63%	45.58	4.98%	193.7	46.79	2.46%	2,031.5	46.34	48.01	3.48%
11	44.03	48.23	8.71%	44.59	7.55%	838.6	47.66	1.18%	1,808.3	47.66	47.77	0.23%
12	46.87	47.61	1.56%	44.25	7.06%	699.0	46.95	1.39%	2,720.6	46.96	47.45	1.03%
13	44.51	47.03	5.35%	44.97	4.38%	293.1	46.40	1.34%	1,816.3	45.21	46.96	3.73%
14	48.45	48.80	0.72%	46.15	5.43%	588.0	48.63	0.35%	2,945.1	47.73	49.24	3.07%
15	41.25	46.26	10.83%	42.95	7.16%	719.3	45.33	2.01%	1,696.0	43.71	46.22	5.43%
16	43.82	47.32	7.39%	45.46	3.93%	298.8	46.47	1.80%	2,087.4	45.97	47.11	2.42%
17	46.20	47.76	3.26%	45.83	4.04%	147.9	45.94	3.81%	1,178.1	46.83	47.73	1.89%
18	49.89	50.33	0.88%	48.61	3.42%	180.0	49.89	0.87%	2,173.4	49.89	50.38	0.97%
19	47.60	48.35	1.55%	46.44	3.95%	268.2	46.95	2.90%	2,248.5	47.6	48.52	1.90%
20	44.91	45.72	1.78%	44.20	3.32%	178.6	44.43	2.82%	2,009.6	44.53	45.92	3.03%
21	38.81	48.12	19.35%	45.78	4.86%	319.8	45.78	4.86%	1,063.3	45.94	48.13	4.55%
22	44.93	54.19	17.09%	49.44	8.77%	221.2	50.56	6.70%	674.7	51.26	52.66	2.66%
23	47.97	49.56	3.21%	47.12	4.92%	210.9	48.26	2.62%	660.9	48.26	49.70	2.90%
24	44.86	46.32	3.16%	44.52	3.89%	184.0	45.04	2.76%	1,156.1	45.83	46.11	0.61%
25	37.05	47.78	22.46%	45.58	4.60%	259.9	45.58	4.60%	806.3	46.65	47.56	1.91%
Avg	46.10	48.80	5.54%	46.58	4.55%	287.0	47.63	2.40%	1,898.6	47.70	48.74	2.12%

Table 3.3: Computational results of the proposed approaches.

which uses the best partition solution (BPS) as an initial solution for IP3.1.

3.5.2.1 Performance of the Global Approach:

We set the runtimes limit of IP3.1 as one hour. In Table 3.3, the index of an instance is reported in the "ID" column. The first major column "Global Approach" contains the results of the global approach. The "Prize" column gives the objective of IP3.1 and the "UB" column gives the upper bound of prizes collected, as found by Gurobi when solving IP3.1. The optimality gap of the IP3.1 solution is given by $100(b-a)/b\%$, where a is the objective of IP3.1 and b is the upper



(a) (b)

Figure 3.8: Partition of our region of interest. Panel (a) shows the default partition. Panel (b) shows the partition when the angle of rotation is $3\pi/15$.

bound found by Gurobi. The "Gap" column presents the optimality gap of the IP3.1 solution. In the last row of Table 3.3, we give the averages of columns except for the "ID" column.

We observe that none of the 25 instances are solved optimally in a 1-hour runtime. The optimality gap ranges from 0.72% to 22.46% with an average of 5.54%. In other words, the optimality gap has a large variance. This is rather undesirable because the optimality gap can be as large as 22.46%. As will become evident later, the other proposed approaches perform better in this regard.

3.5.2.2 Performance of the Partition Approach:

We first discuss our partitioning of the region. Using the center of the HCN as the origin and the negative y-axis as one of the boundaries, our default partition is shown in Figure 3.8a. Because

$K = 3$, the angle between boundaries is $2\pi/3$. By rotating the boundaries in increments of $\pi/15$ in the counterclockwise direction, we have a total of 10 partitions. In Figure 3.8b, we show the partition when the angle of rotation is $3\pi/15$. To make the partition approach more efficient, we implement the following preprocess requirements. First, when a subregion in a partition has more than 40 nodes or fewer than 10 nodes, we disregard the partition, because there is a large imbalance between subregions in terms of the number of nodes. Second, if a partition causes a subregion to only have inner nodes or outer nodes, we disregard the partition, because IP3.2 would be infeasible. For a feasible partition, we then apply IP3.2 for $\theta \in \{70, 73, \ldots, 97, 100\}$ in each subregion. The runtime limit of IP3.2 is 60 seconds. The default optimality gap tolerance (10^{-4}) is applied. After solving IP3.2 for all subregions, we run IP3.3 to maximize the prize. Overall, the partition approach will generate at most 10 solutions.

The second major column "Partition Approach" in Table 3.3 contains the results of the partition approach. The "Prize" column gives the prize collected by the BPS. The "RT(s)" column gives the runtimes in seconds of the partition approach. The "Gap" column presents the optimality gap of the BPS. While the partition approach does not provide valid upper bounds, we use the upper bounds obtained by the global approach to evaluate the optimality gap of the BPS. We observe that the optimality gap of the BPS ranges from 2.07% to 8.77% with an average of 4.55%. Compared to the global approach, the average gap is reduced by 18%, and the maximum gap is significantly reduced by over 60%. More impressively, the average runtime of the partition approach is less than 5 minutes (287.0 seconds), which is over one order of magnitude faster than IP3.1. Thus, the partition approach not only can find solutions of better quality but also is significantly more efficient than the global approach.

3.5.2.3 Performance of the CG Approach:

In the CG approach, we use the partition solutions as the initial columns. We have two enhancements: DOIs and modeling the pricing problem as the ESPPRC. When the pricing problem is solved as an ESPPRC, we pick the source and sink using a fixed or random order of inner nodes. Moreover, we use the bidirectional labeling algorithm with dynamic halfway point in the CSPY package [55] to find the routing solution. If the resulting tour has a reduced cost greater than the tolerance of the reduced cost, we add the tour to $\mathcal{R}'$ and continue to the next CG iteration. Otherwise, we pick another inner node and solve the ESPPRC again. CG is terminated if solving the ESPPRC for all inner nodes yields a tour with a reduced cost smaller than the tolerance of the reduced cost.

Before presenting the performance of the CG approach, we evaluate the impact of the enhancements in the CG approach. To this purpose, we introduce four settings of the CG approach. We start with the one called "IP", for which we solve the pricing problem using IP3.5. The IP3.5 tour is added to $\mathcal{R}'$ if its reduced cost is larger than the tolerance of the reduced cost. Otherwise, CG is terminated. The runtime of IP3.5 in each iteration is limited to 10 minutes. The tolerance of the optimality gap of IP3.5 is set at 5%. The second setting is called "CSPY & Normal", which solves the pricing problem as an ESPPRC with the CSPY package. In addition, we pick the source and sink in the ESPPRC using a fixed order of inner nodes. The next setting is "CSPY & Random", in which, unlike "CSPY & Normal", the source and sink in the ESPPRC are picked randomly. The last setting is "CSPY & Random & DOI". In this case, DOIs are applied in each CG iteration.

We run each setting for a maximum of 3,000 seconds. The tolerance of the reduced cost is

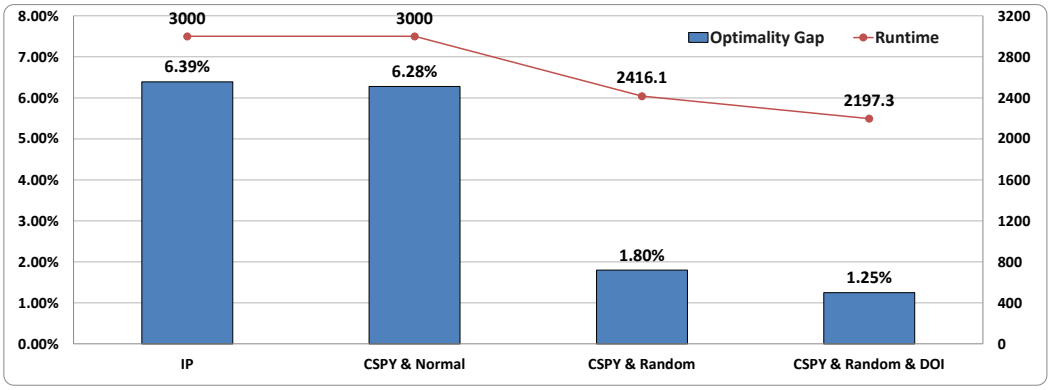
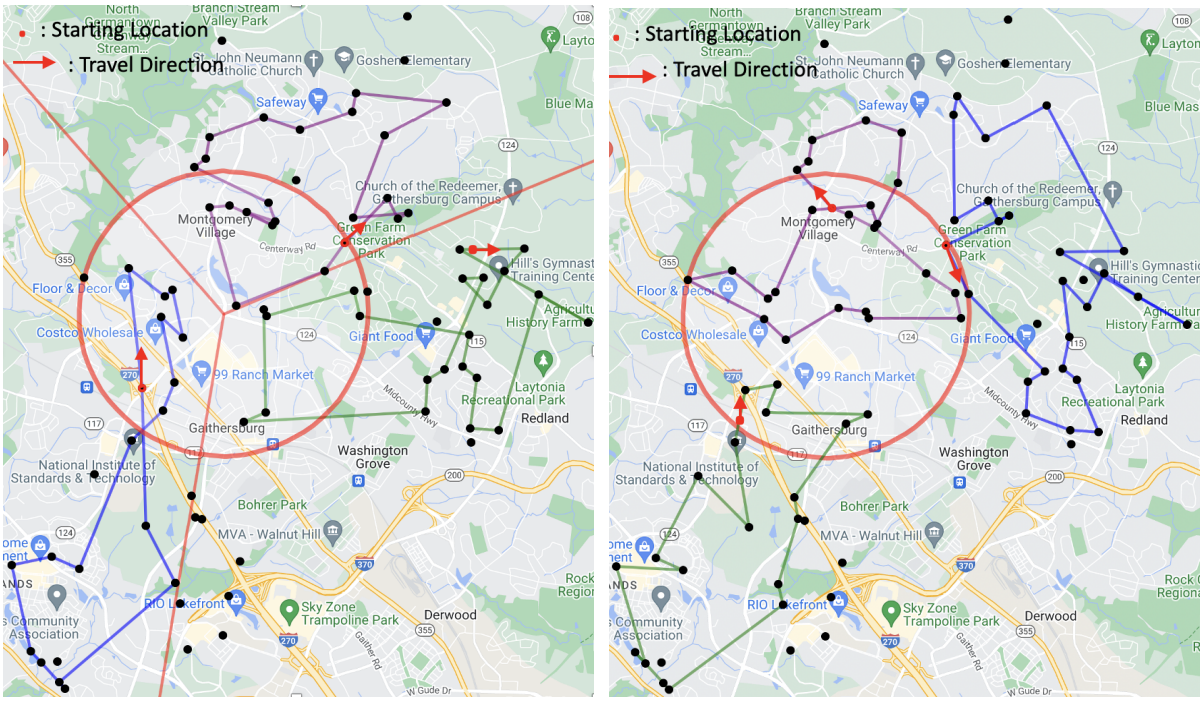


Figure 3.9: Analysis of enhancements in the CG approach. The left y-axis measures the optimality gap. The right y-axis measures the runtimes in seconds.

set to 10^{-6}. After CG is terminated, we run IP3.4R with the updated $\mathcal{R}'$ to find the CG solution. There are two metrics to measure the effectiveness of a setting. The first one is the optimality gap of the solution obtained by each setting, where the upper bound is obtained from the global approach. The second one is the runtime of each setting. We apply all four settings for five instances, which are instances 11 to 15. In Figure 3.9, we plot the performance of the four settings. Clearly, only using the "IP" or "CSPY & Normal" settings has minimal effect on the optimality gap. The major reduction in the optimality gap happens when we choose the source and sink in the ESPPRC randomly ("CSPY & Random"). In particular, the runtime and optimality gap are reduced by 19.5% and 70.3%, respectively, from "CSPY & Normal". Moreover, using the DOIs, we further reduce the runtime and optimality gap by 9.3% and 30.6%, respectively, compared to "CSPY & Random". From now on, we only present the CG approach in "CSPY & Random & DOI" setting.

In Table 3.3, the third major column "CG Approach" has the results of the CG approach. The "Prize" column gives the prize collected by the CG solution. The "RT" column gives the runtimes in seconds for the CG. The "Gap" column presents the optimality gap of the CG solution. The optimality gap of the CG solution has a smaller range (0.35% to 6.70%) compared to



(a) (b)

Figure 3.10: Patrol tours for instance 11. The red dots indicate the starting points and the red arrows indicate the direction of travel. Panel (a) shows the BPS for instance 11. Panel (b) shows the CG solution for instance 11.

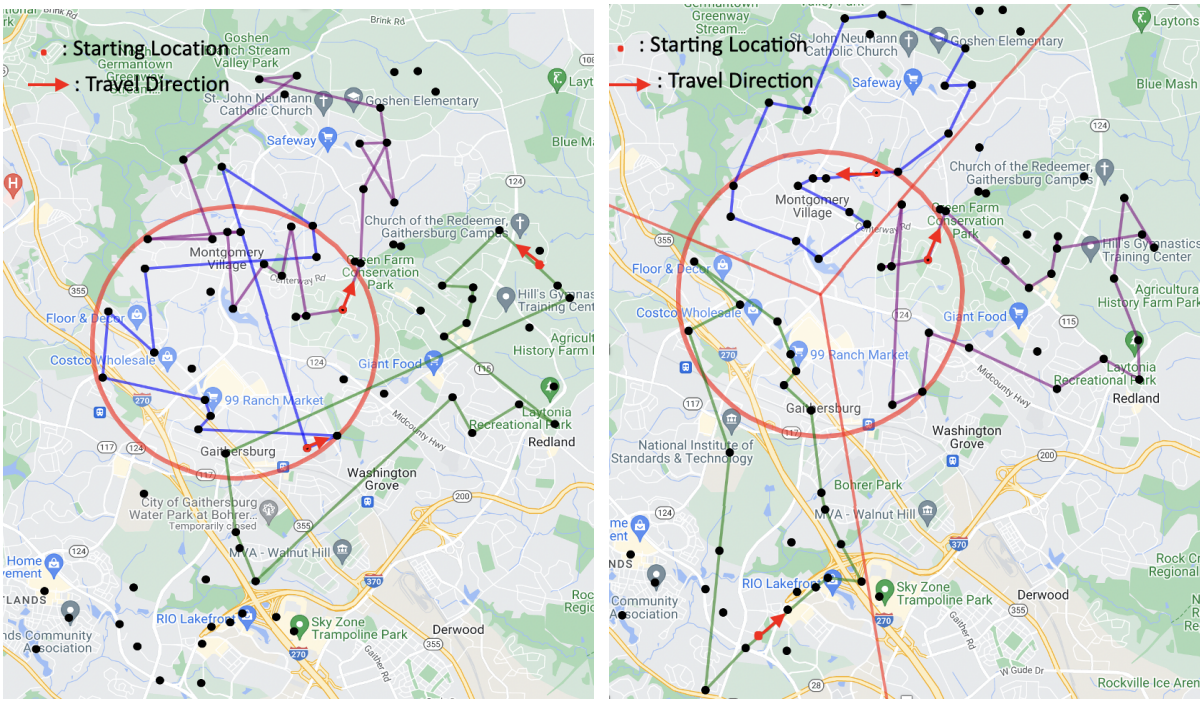
the global and partition approaches. The optimality gap has an average of 2.40%. In addition, the CG approach improves the optimality gap of the partition approach in 21 out of 25 instances. In Figure 3.10, we plot the BPS and CG solutions for instance 11. Note that the arcs in Figure 3.10 indicate the order of visits instead of the actual patrol path. The BPS is suboptimal here because the subregion with the green tour has two clusters of nodes based on the geographical locations. One cluster is located near the boundary shared with the subregion of the purple tour. The other cluster is located near the boundary shared with the subregion of the blue tour. Only one cluster can be covered due to the constraint on the length of the tour. On the other hand, the tours in the CG solution do not have the same problem because these tours are global and not restricted to any partition.

To sum, we conclude that the CG is a good complement to the partition approach because the CG solution improves the average optimality gap of the partition approach by 47%. Moreover, the maximum gap is reduced by 24%. While its runtime is longer compared to the partition approach, the CG approach is able to find better solutions.

3.5.2.4 Performance of the Global Approach with Warm Start:

The CG approach performs well after we include the randomness of selecting the source and sink in the ESPPRC. The downside of this randomness lies in the instability of the CG solution, i.e., each run of the CG could yield a different CG solution. In the column "CG Approach" of Table 3.3, for instances 1, 3, 5, 21, and 25, we see little or no improvement from the CG solution. Due to the randomness, we cannot be sure of the effectiveness of the CG for these instances. To get more stable solutions, we observe that our partition approach has fast runtimes (287.0 seconds) and a reasonable optimality gap (4.55%). By providing the BPS as an initial solution for IP3.1, we give the global approach a warm start and reduce the variance of the optimality gap in IP3.1.

To fairly compare the runtimes of CG and global approach with a warm start, we restrict the runtime of IP3.1 to 3,000 seconds. The last major column in Table 3.3 records the results. The prizes collected in the warm start are shown in "Prize". The "UB" column gives the upper bound of prizes collected, as found by Gurobi, with the warm start. The "Gap" column gives the optimality gap of the IP3.1 solution when using the warm start. We observe that the optimality gap ranges from 0.23% to 5.43%, with an average of 2.12%. The global approach with a warm start further reduces the optimality gap by 12% compared to the CG approach. However, none of



(a) (b)

Figure 3.11: Patrol tours for instance 25. The red dots indicate the starting points and the red arrows indicate the direction of travel. Panel (a) shows the tours of Solution 1 for instance 25. Panel (b) shows the tours of Solution 2 for instance 25.

the 25 instances are solved optimally within 3,000 seconds. As a result, this approach takes 18 minutes longer on average compared to the CG approach. We recommend the global approach with a warm start if the optimality gap is a more important factor than runtime.

3.5.3 Workload Balancing in the HCN

Workload balancing is an important issue receiving more and more attention recently [43]. Especially, given the higher risk of patrolling in the HCN, it is critical to incorporate this factor into consideration. To better visualize the situation, we plot two solution tours of instance 25 obtained by different approaches.

In Figure 3.11a, we plot a solution for instance 25, called Solution 1. We find that there

	Solution 1			Solution 2		
Tour	L	Prize	TW-HCN	L	Prize	TW-HCN
Blue	18.4	11.76	[0, 2.69]	17.7	15.97	[0, 1.71]
Green	19.8	9.00	[2.69, 2.88]	20.0	15.39	[1.71, 2.76]
Purple	20.5	16.28	[2.31, 4]	20.7	14.22	[2.65, 4]

Table 3.4: Length, prize, and time window in the HCN of two solutions of instance 25.

are many unvisited nodes in the southwest corner. Moreover, in the blue and green tours, some edges are quite long. This is not ideal because one prefers patrol cars to collect as many prizes as possible per unit mile. When considering the sequence of covering the HCN, we let the patrol car patrolling the blue tour go first, followed by the patrol car patrolling the green tour and ending with the patrol car patrolling the purple tour. In the first major column "Solution 1" of Table 3.4, we give the length, prizes collected, and time window in the HCN of the tours shown in Figure 3.11a. In particular, the "Tour" column specifies the tour by its color. The "L" column gives the length of each tour in miles. The "Prize" column gives the prizes collected of each tour. The "TW-HCN" (i.e., the time window in the HCN) specifies the time window when each patrol car stays in the HCN. We see that the patrol car covering the blue tour stays in the HCN for 2.69 hours (63% of the patrol time in the HCN), which is much longer than for the other two patrollers. This may not be fair for the patroller because the HCN is usually more dangerous.

In Figure 3.11b, we plot another solution for instance 25, called Solution 2. We observe that more nodes are visited compared to Solution 1. In the second major column "Solution 2" of Table 3.4, we give the length, prizes collected, and time window in the HCN for Solution 2 of instance 25. Compared to Solution 1, more prizes are collected (45.58 in Solution 2 versus 37.05 in Solution 1) and fewer miles are traveled (58.4 in Solution 2 versus 58.7 in Solution 1). The

ID	Partition Approach	CG Approach	Global with Warm Start
1	0.59	0.59	0.26
2	0.52	0.30	0.51
3	0.50	0.50	0.11
4	0.52	0.25	0.06
5	0.58	0.61	0.25
6	0.91	0.07	0.42
7	0.19	0.15	0.23
8	0.12	1.00	0.49
9	0.49	0.23	0.23
10	0.40	1.03	0.25
11	0.35	1.58	1.56
12	0.31	1.58	1.58
13	0.45	1.28	0.35
14	0.24	1.33	1.22
15	0.25	1.56	0.23
16	0.11	0.48	0.11
17	0.39	0.49	0.76
18	0.12	0.32	0.32
19	0.61	0.60	0.71
20	0.20	0.20	1.17
21	0.24	0.24	0.24
22	0.29	0.06	0.53
23	0.36	2.09	2.09
24	0.39	2.10	2.16
25	0.33	0.44	0.41
Avg	0.38	0.76	0.65

Table 3.5: Standard deviations of time in the HCN.

time spent in the HCN is also more balanced in Solution 2. The standard deviation of the time spent in the HCN of three tours is reduced from 1.26 in Solution 1 to 0.33 in Solution 2.

To better understand workload balancing in the HCN, we compute the standard deviation of the time spent in the HCN of the three tours obtained by the following approaches: the partition approach, CG approach, and global approach with a warm start. The results are shown in Table 3.5. The partition approach achieves the best workload balance because its standard deviation is the smallest on average at 0.38. The CG approach leads to solutions with the highest

variance regarding the workload in the HCN. The average of its standard deviation is 0.76. The global approach with a warm start shows better results as its standard deviation averages at 0.65.

3.6 Conclusion

The HSCPP centers on the patrol routing for a fleet of patrol cars in a region with a HCN. We require at least one patrol car to be in the HCN at any given time during the patrol, which might help to reduce the crime rate and response time in the HCN. We develop several solution approaches to find the patrol routes. To determine the starting points of the routes, we present a simple and efficient search algorithm. Using simulated instances based on real crime data in Montgomery County, Maryland, we evaluate the effectiveness of our solution approaches based on the optimality gap and runtimes. For the global approach, although the routing solutions for all patrol cars can be found using a single IP formulation, the optimality gap has a large variance with a 1-hour runtime. For the partition approach, high-quality solutions with an optimality gap of 4.55% can be found within 5 minutes of runtime. Using the partition solutions as the initial columns for the CG approach, the optimality gap can be reduced to 2.4% with 36 minutes of runtime. To speed up the convergence of the CG approach, we implement DOIs and model the pricing problem as an ESPPRC. Lastly, we consider using the BPS as a warm start for the global approach. In this case, the optimality gap decreases to 2.12% with 55 minutes of runtime. We also study workload balancing in the HCN and find that the partition approach clearly outperforms the CG approach and the global approach with a warm start. Overall, we recommend using our partition approach if workload balancing and runtime are the priorities. Alternatively, we recommend using the global approach with a warm start because this approach does well with

respect to the optimality gap and workload balancing.

Future work can explore different ways in which a region can be partitioned. For instance, it may make sense to partition the region based on the number of prizes or the number of nodes. Moreover, we hope to design efficient heuristics for the pricing problem of column generation to handle an even larger number of patrol locations. We are also interested in introducing randomness into the patrol routes so that the routes can be unpredictable. For example, we could increase the prizes associated with uncovered locations after each patrol and re-optimize routes repeatedly.

Chapter 4: The Paired Mail Carrier Problem

4.1 Introduction

Several national postal services are looking to reduce the number of trucks required to service their customers. The primary motivation for these changes is the desire to reduce capital and truck maintenance costs. With fewer trucks, parking becomes less of a problem, especially when delivering in urban areas [21, 30, 50]. Furthermore, postal trucks stop frequently along their routes, so they cause more pollution and obstruction than moving automobiles. Therefore, a smaller number of trucks results in the reduction of carbon footprints and traffic, especially in the city [19].

Recently, novel operations models have been proposed to reduce the truck fleet size and emissions footprint for a company. One way is to use autonomous vehicles [51, 52]. An autonomous vehicle can drop a carrier to make deliveries and pick up the carrier at a different location when the deliveries have been completed. In addition, the carrier can sort mail while the autonomous vehicle is driving itself. Other attempts have been made to reduce both fleet size and carbon emissions. For example, the University of Washington delivers mail with electric bikes instead of trucks [45]. Deutsche Post DHL Group uses ecofriendly cargo bikes for deliveries in urban cities including New York and Miami [24]. UPS places mobile distribution centers near the city center and delivers packages by electric bike or foot [62]. The United States Postal Service

(USPS) has carriers drive to neighborhoods and then deliver mail on foot to reduce emissions and save fuel [63]. Royal Mail proposes to use more than one mail carrier per truck [27].

In this chapter, we explore the potential benefits of using more than one mail carrier per truck. We allow a mail carrier to travel between stops independently of the other mail carriers. In particular, mail carriers can walk or drive between stops. If light vehicles such as bicycles are equipped in the truck, walking can be replaced by using the light vehicles. Moreover, we focus on the case of two mail carriers per truck for two reasons. Delivery trucks usually have at most two seats for mail carriers (e.g., trucks used by the USPS). Moreover, although there are potential benefits from using more than two mail carriers per truck, the marginal benefit of adding an extra mail carrier decreases, as we will demonstrate in Section 4.5.

In the literature, using more than one mail carrier per truck is modeled as a variant of the vehicle routing problem (VRP) [1, 49, 56]. The objective of the variant is to minimize a weighted linear sum of three components: the fleet size (number of trucks), number of delivery workers, and total distance traveled by the trucks. Customers are clustered around available parking spaces, and the service time of a cluster is a known function of the number of service workers on the truck assigned to the cluster. Cluster assignments, the aggregate demand at each cluster, and the service time function are treated as problem inputs. The number of trucks, routes of the trucks, and number of workers assigned to each truck are decision variables. When solving this problem, methods such as ant colony optimization, tabu search, and mathematical programming are applied in Alvarez and Munari [1], Senarclens de Granc and Reimann [56], and Pureza et al. [49].

The ability for a carrier to travel between stops independently has been shown to be advantageous in the literature. McLeod et al. [44] consider the delivery model, where the truck brings

the mail and packages to collection points, then carriers deliver them from the collection points by walking, riding an electric bike, or riding an electric quadcycle. Their results show that carbon emissions and driving distance are significantly reduced when using this novel model. Fikar and Hirsch [25] consider the routing and scheduling problem of transporting nurses for home health care services and incorporate walking between assignments for nurses. Their results show a substantial reduction in the number of required vehicles for transporting the nurses. Coindreau et al. [18] consider the VRP of sending technicians for on-site services and allow walking and carpooling for technicians to reach a job. Their results show significant savings in the fleet size and distance traveled by vehicles. Lin [40] considers a pick up and delivery problem for a set of locations involving a truck with multiple carriers. Carriers can pick up and deliver by foot or using the truck. The author shows that by allowing the carriers to work independently, a lower total cost can be achieved.

We study a problem called the Paired Mail Carrier Problem (PMCP). For the setup in the PMCP, we have a truck making several stops along a mail route denoted as a truck route. A stop may have multiple customers, and we treat it as a single predefined block of work with a timespan. In the traditional single mail carrier (SMC) setting, the mail carrier drives the truck following the truck route and exits the truck at each stop to service customers. When finished, the mail carrier returns to the truck and drives to the next stop. In the paired mail carrier (PMC) setting, the work allocation between mail carriers makes the problem much more interesting. Specifically, when one mail carrier is servicing stop j, the other mail carrier has the option of driving the truck or walk to the next unserviced stop $j+1$ or waiting at stop j. Using the first two options, fewer unserviced stops are left on the truck route. Moreover, by walking, the truck is left to the mail carrier who is servicing stop j and that helps the trailing carrier to catch up and

service future stops on the truck route. Our objective in the PMCP is to minimize the completion time for servicing all stops, subject to the following feasibility constraints. First, the truck and all mail carriers must reach the end of the route. Second, the mail carriers and the truck can only move forward along the truck route. Third, a mail carrier must be in the truck for the truck to move forward. The outputs of our PMCP include the allocation of work between mail carriers, how each mail carrier travels between stops, and where the truck is parked along the truck route.

The main contributions of this chapter are as follows: First, we develop three solution approaches, a mixed integer programming (MIP) formulation and two fast heuristics, for the PMCP and conduct extensive computational experiments. The MIP formulation solves the PMCP exactly. For smaller instances, it obtains optimal solutions within reasonable running times (a few seconds at most). Further, our heuristics are extremely efficient. Their running times are on the order of milliseconds even for larger instances. In addition, we show that the one of our more sophisticated heuristics can find near-optimal solutions on instances up to 80 service stops. The optimality gap is within 5%.

More importantly, we evaluate the impact of the PMC setting on both a one-truck situation and a fleet (multiple trucks) situation, relative to the SMC setting. On the one hand, considering one truck, the PMC setting can accomplish over 50% extra work within the same shift hours. On the other hand, considering a fleet of trucks, the PMC setting can lead to 22% cost savings. Overall, we demonstrate that the PMC setting as a novel operation mode is worth considering in practice. Lastly, we discuss the case when there are k mail carriers per truck, where k can be any positive integer number. We present an MIP formulation and extend one of our heuristics for this general situation.

The remainder of this chapter is structured as follows. In Section 4.2, we define the PMCP

mathematically and introduce several propositions to narrow the search space for our solution approaches. In Section 4.3, we present the MIP formulation and two heuristics. In Section 4.4, computational experiments are performed to evaluate the impact of the PMC setting based on the solutions of the MIP formulation. Further, we discuss the performance of the MIP formulations and the heuristics. Section 4.5 discusses the extension to k mail carriers per truck. Finally, concluding remarks are given in Section 4.6.

4.2 Definition and Dominated Actions of the PMCP

In this section, we formally define the PMCP and present several dominated actions that should be avoided in the search process for high-quality solutions.

In the PMCP, we consider a truck route T with N stops. The nonstop truck travel time for truck route T is L minutes. Assume that the information about the location of a stop and service time of a stop are known and fixed. This information is known, given that postal deliveries typically follow a fixed route with the same set of customers each day. Servicing a stop is a closed walking loop (see Figure 4.1). The walking loop cannot be split between mail carriers, i.e., if the service time of a stop is t minutes, one mail carrier must spend t uninterrupted minutes servicing the stop. In addition, we denote the service time of stop j and the nonstop truck travel time between the start of truck route T and stop j by s_j and d_j, respectively. We measure the location of mail carrier A on the truck route by the nonstop truck travel time between the location of A and the start of truck route T. In particular, the location on the truck route of mail carrier A at time t is denoted as $loc(A, t)$. Similarly, $loc(B, t)$ and $loc(\mathcal{T}, t)$ denote the locations of mail carrier B and the truck at time t, respectively. We denote the ratio of the truck speed to the walk

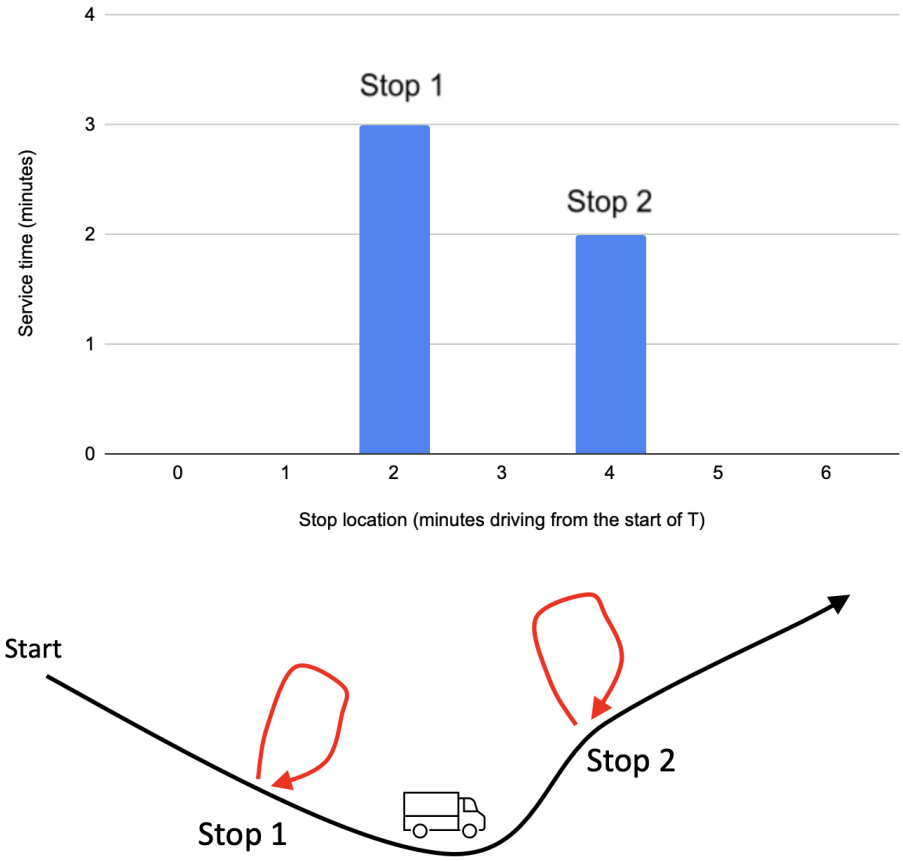
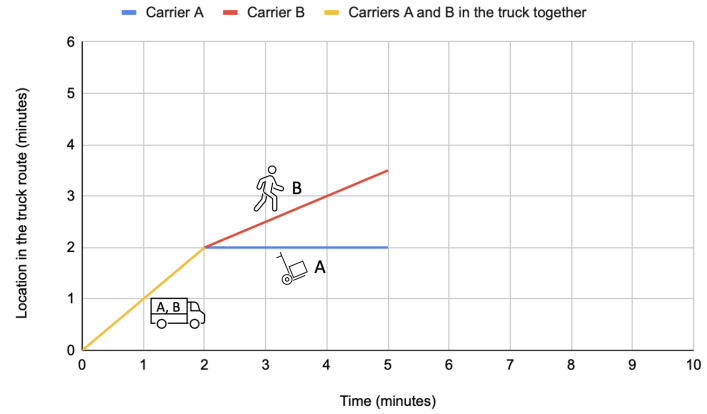


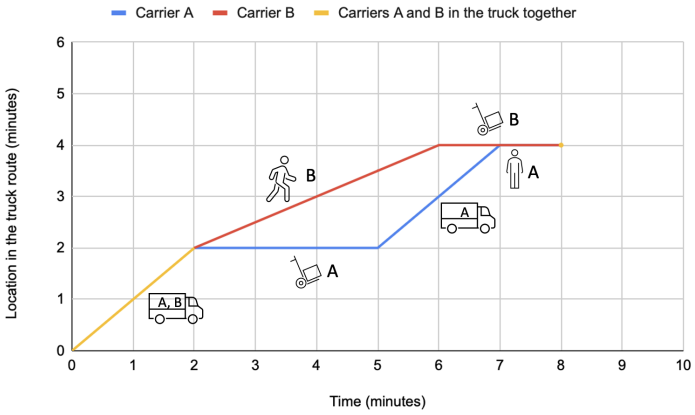
Figure 4.1: In the top image, we plot the length of the service time at a stop based on the stop location on the truck route. In the bottom image, we show the truck route in black. Stops are serviced using the closed walking loops in red.

speed along T by α. The value of α is greater than 1, given that walking is slower than driving the truck. For example, walking from the start of truck route T to stop 1 requires αd_1 minutes. We use α to model the traffic condition of the route. As α decreases to one, the traffic conditions worsen.

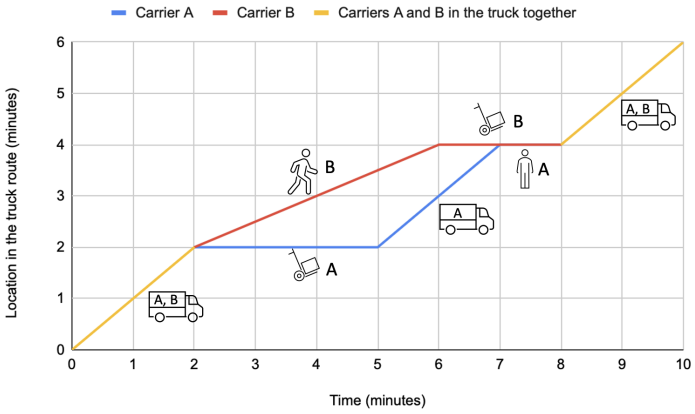
In Figure 4.1, we present an example of the PMCP. In the top image, we plot the length of the service time at a stop based on the stop location on the truck route. In the bottom image, we show the truck route in black. Stops are serviced using the closed walking loops in red. In this example, $L = 6$, $N = 2$, $s_1 = 3$, $d_1 = 2$, $s_2 = 2$, and $d_2 = 4$. For instance, s_1=3 indicates that it



(a)



(b)



(c)

Figure 4.2: The trajectories of mail carriers A and B and the truck versus time in a solution of the example shown in Figure 4.1. Each panel shows a snapshot of the trajectories over a time interval.

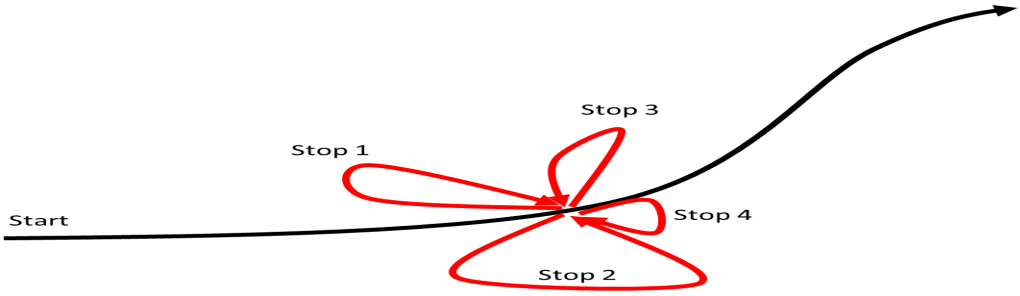


Figure 4.3: A special case of the PMCP with $d_1 = d_2 = d_3 = d_4 = 5$.

takes three minutes to service stop 1. $d_2 = 4$ indicates that it takes four minutes for a mail carrier to drive a truck from the start to stop 2 (without stopping). We set $\alpha = 2$ and illustrate a solution of this example in Figure 4.2 by utilizing a time-space diagram. The x-axis measures the time. The y-axis measures the locations of the mail carriers and truck. In the solution, we have both carriers riding in the truck from the start to stop 1. The arrival time at stop 1 is $t = 2$ minutes. Then, mail carrier A is assigned to service stop 1. Mail carrier B walks to stop 2 (see Figure 4.2a). Mail carrier B arrives at stop 2 at $t = 6$ minutes and starts servicing stop 2. Mail carrier A finishes servicing stop 1 at $t = 5$ minutes and drives the truck to reach stop 2 at $t = 7$ minutes. Mail carrier A idles at stop 2 to wait for mail carrier B (see Figure 4.2b). At $t = 8$ minutes, both carriers use the truck to travel to the end of truck route T (see Figure 4.2c). The completion time of the solution is 10 minutes.

Before proceeding, we show that the PMCP is NP-hard. The reduction is done through the partitioning problem, which is NP-hard [35]. The partition problem involves partitioning a given multiset S of positive integers into two subsets S_1 and S_2 such that the difference between the sum of elements in S_1 and the sum of elements in S_2 is minimized. Consider a special case of the PMCP in which all service stops are at the same location along the truck route. Figure 4.3

shows an instance of this special PMCP case. The optimal solution of this special case of the PMCP balances the total service time as evenly as possible among mail carriers. Moreover, this optimal solution solves the partitioning problem where the service time of a stop is a number in the partition problem. Therefore, the PMCP generalizes the partitioning problem.

4.2.1 Dominated Actions

We introduce several propositions to narrow the search space for our solution approaches. These propositions show that certain actions should be avoided, given that a better alternative exists and yields a shorter completion time (the objective value). We call these actions as dominated actions. They become instrumental later when we develop our solution approaches.

We define several terms first. If $loc(A,t) > loc(B,t)$, then A is the leading mail carrier, and B is the trailing mail carrier at time t. If $loc(A,t) = loc(B,t)$, B is busy servicing, and A is idle (and therefore, free to move forward), then A is the leading mail carrier, and B is the trailing mail carrier at time t.

Regarding possible actions, when mail carrier A is idle and intends to move forward, A could (i) walk forward, (ii) drive the truck forward if the truck is with A, or (iii) wait and move forward using the truck with mail carrier B if A is the leading mail carrier. For case (iii), we say that mail carrier A has fully waited for B. If mail carrier A waits but not until B rejoins, we say that A has partially waited.

Proposition 3 *An idle mail carrier should never wait partially.*

Proof: Included in Appendix C.

Proposition 4 *It is never advantageous for a leading mail carrier to walk partially, i.e., for the leading mail carrier to walk only part of the way from stops j to $j+1$, then wait to be picked up by the trailing one in the truck.*

Proof: Included in Appendix C.

Proposition 5 *There exists an optimal solution such that if a walking mail carrier gets into a non-empty truck (i.e., the other mail carrier is in the truck.), it happens at a stop.*

Proof: Included in Appendix C.

Proposition 6 *In a solution of the PMCP, the truck is always between the two mail carriers, i.e.,* $\min(loc(A,t), loc(B,t)) \leq loc(\mathcal{T},t) \leq \max(loc(A,t), loc(B,t))$ *for any* t.

Proof: Included in Appendix C.

4.3 Solution Approaches of the PMCP

We develop solution approaches in this section. In Section 4.3.1, we derive an MIP formulation, which yields an exact solution for the PMCP. In Section 4.3.2, we present two heuristics. It is important to note that our PMCP can be modeled as a machine scheduling problem. Using the standard 3-field notation, the PMCP is classified as the $P2||C_{\max}$ problem, which can be solved by dynamic programming (DP). We propose the DP formulation in Appendix D. We find that the DP formulation has infinitely many states and actions because an idle leading carrier at stop j can drive to any location between stops j and $j+1$, park the truck, and walk to stop $j+1$. Each parking location corresponds to a different arrival time at stop $j+1$ and leads to a unique state in the DP. Driving to this parking location and then walking to stop $j+1$ counts as an action in the

DP. Since parking locations are uncountable, the states and actions are both infinite. We suspect that the DP might be suboptimal in terms of runtime due to infinite states and actions. Therefore, designing efficient algorithms to solve the DP formulation is left as a potential future research direction.

4.3.1 MIP Formulation

For our MIP, we let stop 0 and stop $N+1$ be the start and the end points of the truck route, respectively. Both have a service time of 0. It follows that $d_0 = s_0 = s_{N+1} = 0$ and $d_{N+1} = L$. $I = \{A, B\}$ is the set of mail carriers, $J = \{0, 1, 2, \ldots, N, N+1\}$ is the union of service stops, stop 0, and stop $N+1$. $J^- = \{0, 1, 2, \ldots, N\}$ is the set of J except the end point. Let i be the index of I and j be that of J and J^-. We denote M by the "big M" commonly used as a large constant in MIP contexts. For the MIP variables, ct is the completion time, $A_{i,j}$ is the arrival time of mail carrier i at stop j, $D_{i,j}$ is the departure time of mail carrier i at stop j, and $tt_{i,j} \in [0, 1]$ is the portion of travel from stops j to $j+1$ spent in the truck by mail carrier i. For example, if $tt_{B,12} = 0.25$, then mail carrier B traveled in the truck for one-quarter of the distance from stops 12 to 13. Let $x_{i,j} = 1$ if mail carrier i services stop j; otherwise $x_{i,j} = 0$. Similarly, let $b_j = 1$ if both mail carriers leave stop j in the truck; otherwise, $b_j = 0$. Our MIP formulation, denoted by

MIP4.1, is given below.

$$\text{(MIP4.1) Min} \quad ct \tag{4.1}$$

$$\text{Subject to} \quad \sum_{i \in I} x_{i,j} = 1 \qquad \forall j \in J \tag{4.2}$$

$$A_{i,j+1} \geq D_{i,j} + (d_{j+1} - d_j)(tt_{i,j} + \alpha(1 - tt_{i,j})) \qquad \forall i \in I, j \in J^- \tag{4.3}$$

$$D_{i,j} \geq A_{i,j} + s_j x_{i,j} \qquad \forall i \in I, j \in J \tag{4.4}$$

$$D_{A,j} - D_{B,j} \geq -M(1 - b_j) \qquad \forall j \in J \tag{4.5}$$

$$D_{A,j} - D_{B,j} \leq M(1 - b_j) \qquad \forall j \in J \tag{4.6}$$

$$\sum_{i \in I} tt_{i,j} = 1 + b_j \qquad \forall j \in J^- \tag{4.7}$$

$$ct \geq A_{i,N+1} \qquad \forall i \in I \tag{4.8}$$

$$A_{i,0} = 0 \qquad \forall i \in I \tag{4.9}$$

$$A_{i,j}, D_{i,j} \geq 0 \qquad \forall i \in I, j \in J \tag{4.10}$$

$$0 \leq tt_{i,j} \leq 1 \qquad \forall i \in I, j \in J^- \tag{4.11}$$

$$x_{i,j} \in \{0, 1\} \qquad \forall i \in I, j \in J \tag{4.12}$$

$$b_j \in \{0, 1\} \qquad \forall j \in J \tag{4.13}$$

In MIP4.1, the objective function (4.1) minimizes the completion time, which is the objective of the PMCP. Constraints (4.2) require each stop to be serviced by exactly one mail carrier. Constraints (4.3) force the arrival time at stop $j+1$ of mail carrier i to be the sum of the departure time from stop j of mail carrier i, the amount of time spent walking between stops j and $j+1$ of mail carrier i, and the amount of time spent in the truck between stops j and $j+1$ of mail carrier i. Constraints (4.4) calculate the departure time of mail carrier i at stop j by enforcing the service

time if mail carrier i is assigned to stop j. Constraints (4.5) and (4.6) together posit that if both mail carriers leave stop j at the same time, $b_j = 1$. Constraints (4.7) force Proposition 5, i.e., the mail carriers can only get in the non-empty truck at a stop. Constraints (4.8) make sure that the objective value of the PMCP is greater than or equal to the completion times of the mail carriers. Constraints (4.9) set the starting time as 0. Constraints (4.10), (4.11), (4.12) and (4.13) give the domains of our variables.

Now, we discuss how to set the M value in Constraints (4.5) and (4.6). Although we can easily set a very large value, a smaller value can significantly reduce the running time . At stop j, the largest possible difference between $d_{A,j}$ and $d_{B,j}$ is obtained when one mail carrier always drives the truck, and the other carrier always walks and serves all stops between stops 1 and j. Hence, M can be set as $(\alpha - 1)(d_j - d_1) + \sum_{k=1}^{j} s_k$. We could also make use of the objective value z_H of the balanced heuristic solution, which will be studied below. z_H is an upper bound on the optimal objective value. At stop j, the earliest departure time of a mail carrier is d_j and the latest departure time of a mail carrier is $z_H - (d_{N+1} - d_j)$. Thus, the value of M can be tightened as $z_H - d_{N+1}$. We believe that the value of M can be further reduced, although more effort would be required. In our computational experiments, we set $M = \min\{(\alpha - 1)(d_j - d_1) + \sum_{k=1}^{j} s_k,\ z_H - d_{N+1}\}$ for stop j. As will be evident later, this provides satisfactory performance in computational experiments.

4.3.2 Heuristics

While MIP4.1 performs well on smaller instances, it has difficulties on larger instances. We resort to heuristics to find good-quality solutions efficiently. In this subsection, we introduce

a constructive heuristic that calls the basic heuristic. Afterward, we present the balanced heuristic which builds on the basic heuristic solution and applies a local improvement procedure.

When designing a heuristic for the PMCP, we must consider the actions of an idle leading mail carrier. Assume that A is the idle leading mail carrier at stop j, with $1 \leq j \leq N$. Using Propositions 1-4, we observe that there are four feasible actions for A that are not temporally dominated by other actions.

Action 1: Wait until the trailing mail carrier B arrives, then continue forward together in the truck to stop $j + 1$. Upon arrival, one of the mail carriers is then randomly selected to service the stop.

Action 2: Walk forward to stop $j + 1$ and begin servicing.

Action 3: If the truck is with A, use the truck to travel to stop $j + 1$ and begin servicing.

Action 4: If the truck is with A, use the truck to travel to some location between stops j and $j + 1$, then walk to stop $j + 1$ and begin servicing.

At stop 0 (the start of the route), both mail carriers ride the truck to stop 1. Without loss of generality, we assume that mail carrier A services stop 1. At stop N (the last service stop), Action 1 would be chosen because the objective value of the PMCP equals the larger completion time among mail carriers.

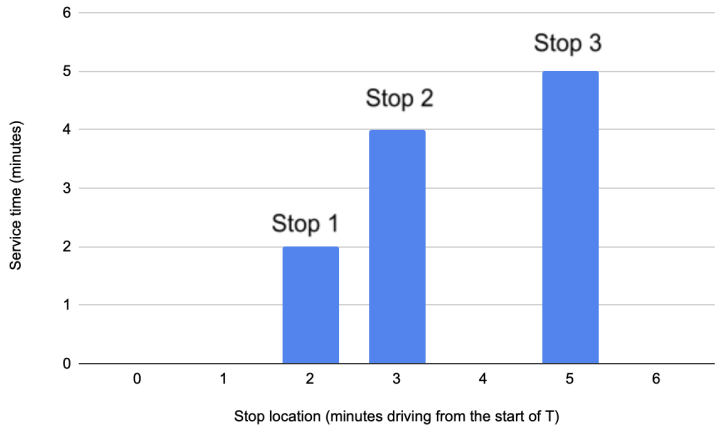
For our basic heuristic, we only consider Actions 1 and 2. The intuition is that if A moves alone in the truck (Actions 3 and 4), the two mail carriers will be farther away from each other at later stops. This would lead to more walking for B. If α is large, this is disadvantageous for the completion time of B and the objective value. For our balanced heuristic, we build from the

solution of the basic heuristic, incorporate Actions 3 and 4 into the solution, and try to improve the objective value.

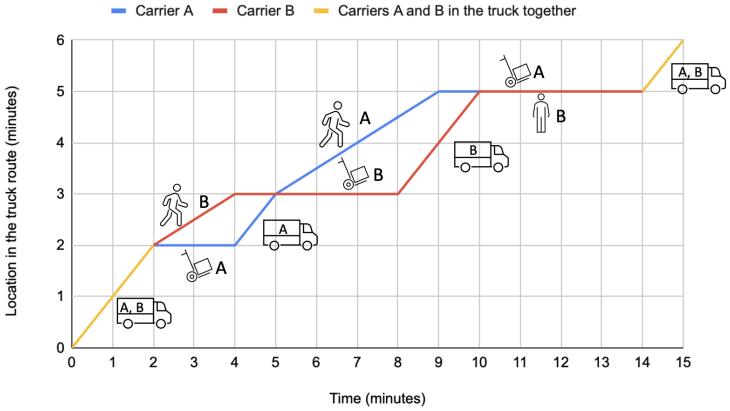
4.3.2.1. Basic Heuristic

For the basic heuristic, we choose between Actions 1 and 2. Suppose that A is the idle leading mail carrier at stop j. We define $waitTime$ as the time in minutes required for the trailing mail carrier B to be ready to depart stop j with A using the truck. If Action 1 is chosen, the time required for A to reach stop $j+1$ is $d_{j+1} - d_j + waitTime$. If Action 2 is chosen, the time required for A to reach stop $j+1$ is $\alpha(d_{j+1} - d_j)$. A acts based on the action that requires the least amount of time to reach stop $j+1$. When both actions yield the same arrival time at stop $j+1$, Action 1 is chosen to save the mail carrier's physical energy. It is important to note that the trailing mail carrier always drives the truck in the basic heuristic solution and will not service any stop until the carrier catches up to the leading mail carrier.

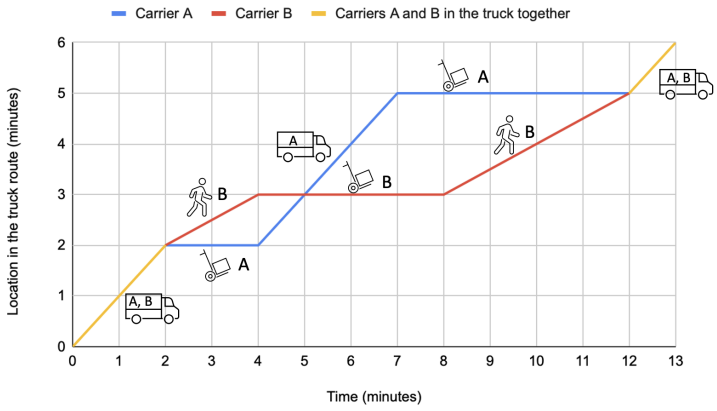
We illustrate the basic heuristic using the example in Figure 4.4a. In this example, $L = 6$, $s_1 = 2$, $d_1 = 2$, $s_2 = 4$, $d_2 = 3$, $s_3 = 5$, and $d_3 = 5$. In Figure 4.4b, we illustrate the solution obtained by the basic heuristic and plot the trajectories of mail carriers A and B and the truck versus time. For $\alpha = 2$, our basic heuristic begins with both mail carriers moving forward in the truck to stop 1 and arriving at $t = 2$ minutes. We assign mail carrier A to begin servicing stop 1, which requires two minutes. As A begins to service, mail carrier B is the idle leading mail carrier. Waiting for A and driving to stop 2 requires $2 + 1 = 3$ minutes. Walking directly to stop 2 would require $\alpha \times 1 = 2 \times 1 = 2$ minutes. As a result, B will choose to walk. Mail carrier B reaches stop 2 at $t = 4$ minutes and starts to service. Mail carrier A finishes servicing stop 1 at $t = 4$ minutes and arrives at stop 2 in the truck at $t = 5$ minutes. Now, A becomes the idle leading mail carrier and decides to walk instead of waiting because walking takes 4 minutes to get to stop



(a) An example of the truck route with three stops.



(b) The trajectories of mail carriers A and B and the truck versus time in the basic heuristic solution.



(c) The trajectories of mail carriers A and B and the truck versus time in the balanced heuristic solution.

Figure 4.4: An example of the truck route with three stops to illustrate the basic and balanced heuristics.

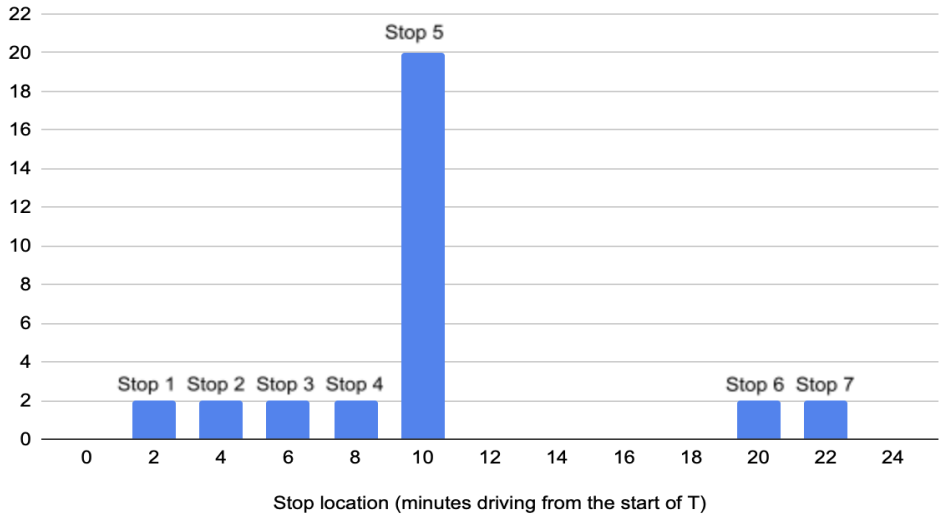


Figure 4.5: An example illustrating the poor performance of our basic heuristic.

3, whereas waiting takes 5 minutes (3 minutes waiting plus 2 minutes driving). Consequently, A reaches stop 3 at $t = 9$ minutes and finishes servicing at $t = 14$ minutes. Mail carrier B gets to stop 3 at $t = 10$ minutes in the truck. Given that there are no more stops on the truck route, both mail carriers ride the truck to the end of the route. The objective value of the basic heuristic solution is 15 minutes.

4.3.2.2. Balanced Heuristic

The basic heuristic performs poorly when both Actions 1 and 2 take a long time to reach the next stop. For example, in Figure 4.5, we have $L = 24$, $s_1 = 2$, $d_1 = 2$, $s_2 = 2$, $d_2 = 4$, $s_3 = 2$, $d_3 = 6$, $s_4 = 2$, $d_4 = 8$, $s_5 = 20$, $d_5 = 10$, $s_6 = 2$, $d_6 = 20$, $s_7 = 2$, and $d_7 = 22$. With $\alpha \geq 2$ in the basic heuristic solution, the idle leading mail carrier chooses to wait at stop 5, remaining unproductive for 20 minutes. With $\alpha < 2$, the idle leading mail carrier chooses to walk to stop 6, but there is a long gap between stops 5 and 6. As a result, both actions are not ideal in terms of the mail carrier's arrival time at stop 6.

To handle the type of situation in Figure 4.5, we must incorporate Actions 3 and 4 (as

described earlier) into the basic heuristic solution. When constructing the balanced heuristic solution, the allocation of service stops remains the same as in the basic heuristic solution. We change how the mail carriers commute between stops to achieve a smaller completion time.

We first segment the basic heuristic solution. A segment starts when both mail carriers arrive together in the truck at a stop j_s and finishes when both mail carriers leave a stop j_f in the truck together. The stop $j_f + 1$ is the starting point of the next segment. We express segment p of the basic heuristic solution as an interval $G_p = [j_{p,s}, j_{p,f}]$. For example, in Figure 4.4a, for $\alpha = 2$, there is one segment in the basic heuristic solution, i.e., $G_1 = [1, 3]$. For a segment G_p of the basic heuristic solution, we define the segment completion time of a mail carrier as the time at which the carrier is ready to leave for the next segment. Because the mail carriers could have different segment completion times, the waiting time of the idle leading mail carrier at stop $j_{p,f}$ is the difference in the segment completion time between the mail carriers. Suppose that mail carrier A is the idle leading mail carrier at stop $j_{p,f}$ and the waiting time of A at stop $j_{p,f}$ is g minutes. If we are able to delay A's segment completion time by $g/2$ minutes in exchange for mail carrier B to complete the segment $g/2$ minutes earlier, we would take this exchange. Both mail carriers could depart together in the truck to the next segment $g/2$ minutes earlier. We achieve this by replacing some of B's walking by driving. In particular, $\alpha g/(2(\alpha - 1))$ minutes of walking is replaced by $g/(2(\alpha - 1))$ minutes of driving. Then, A will walk for $\alpha g/(2(\alpha - 1))$ additional minutes.

To determine the location for the exchange, we identify the last stop in the segment encountered by mail carrier B, where B was the idle leading mail carrier, with the truck at some stop j, and B decided to walk to the next stop $j + 1$. If this type of last stop does not exist, the balanced heuristic solution in the segment is the same as the basic heuristic solution. Suppose that this type

of last stop does exist and occurs at stop j. If the travel time of the truck between stops j and $j+1$ is at least $g/(2(\alpha-1))$ minutes, the balanced heuristic solution for the segment has B driving the truck for $g/(2(\alpha-1))$ minutes from stop j, getting out of the truck, and walking to service stop $j+1$. Mail carrier A walks for $\alpha g/(2(\alpha-1))$ minutes from stop j, gets in the truck, and drives to stop $j+1$. We successfully eliminate the waiting time of A at stop $j_{p,f}$. The completion times of the segment and the entire route both decrease by $g/2$ minutes. On the other hand, suppose that the truck travel time from stops j to $j+1$ is less than $g/(2(\alpha-1))$ minutes. Then, the balanced heuristic solution has B driving the truck from stops j to $j+1$ and A walking from stops j to $j+1$. In this case, the waiting time of A at stop $j+1$ reduces by $2(\alpha-1)(d_{j+1}-d_j)$ minutes. The completion times of the segment and the entire route both decrease by $(\alpha-1)(d_{j+1}-d_j)$ minutes.

For the instance in Figure 4.4a, in the basic heuristic solution, mail carrier B waits for mail carrier A at stop 3 (the end of the segment [1, 3]) for four minutes. In Figure 4.4c, we illustrate the solution obtained by the balanced heuristic and plot the trajectories of mail carriers A and B and the truck versus time. In the balanced heuristic, we replace four minutes of A's walking by driving. We backtrack to the last stop, where A was the idle leading mail carrier, with the truck, and decided to walk to the next stop. This is stop 2. The travel time of the truck between stops 2 and 3 is two minutes. In the balanced heuristic solution, A drives the truck to stop 3 immediately without B and arrives at $t = 7$ minutes. Mail carrier A completes servicing stop 3 at $t = 12$ minutes. Mail carrier B walks from stops 2 to 3 and reaches stop 3 at $t = 12$ minutes. Then, they both ride the truck to the end point at $t = 13$ minutes. The completion time of the route decreases by two minutes with the balanced heuristic.

4.4 Computational Experiments

In this section, we perform computational experiments with MIP4.1, the basic heuristic, and the balanced heuristic. Using MIP4.1, we evaluate the impact of the PMC setting relative to the SMC setting in two situations: a single truck and a fleet of trucks. First, we consider a single truck and compute the percentage savings in the completion time of the PMC setting over the SMC setting. We further incorporate the savings into the extra stops that can be accomplished by the PMC setting within the same shift hours of the SMC setting. Second, we consider a fleet of trucks. Specifically, we compare the costs of servicing an area with a fleet of trucks with a single carrier (denoted by SMC fleet), a fleet of trucks with paired carriers (denoted by PMC fleet), and a fleet of trucks with single and paired carriers (denoted by hybrid fleet). In addition, we test the efficacy and efficiency of the two proposed heuristics.

All computations are performed on a computer with an Apple M1 8-core processor and 8 GB of RAM. MIP4.1 is solved by Gurobi 9.5 via Python API with the default setting. The two proposed heuristics are implemented in Python.

4.4.1 The Impact of the PMC Setting on a Truck Route

In this experiment, we generate random truck routes. These instances are on the smaller side, as we want to obtain optimal solutions by MIP4.1. Then, we can evaluate the impact of the PMC setting based on the optimal solutions. There are $N = 20$ service stops along the route. The travel time between adjacent stops is generated by a uniform distribution $\mathcal{U}(0, D)$. The service time for a stop is generated by a truncated normal distribution $\mathcal{N}(\mu, \sigma^2)$, where a service time above 2μ or below 0 is disregarded and a new one is generated. This symmetry preserves μ as the

mean of the distribution and eliminates any complications from negative service times. In order to generalize our experiments, we no longer require D or μ to be measured in minutes. The time units of D and μ can be adapted according to the application. For example, the time unit can be 10 minutes when delivering in rural areas.

To fully evaluate the impact of the PMC setting, we consider two measures. The first one is the percentage savings in the completion time. For each route, we compute the completion time of the PMCP using MIP4.1. The completion time of the SMC solution is just the length of the truck route plus the sum of all service times. The percentage savings generated by the PMC setting over the SMC setting for a particular instance are given by $100((a-b)/a)\%$, where a is the completion time of the SMC solution, and b is the completion time of the PMCP. The second measure is the number of extra stops that can be generated by utilizing the savings in the completion time. To preserve the distribution regarding the distance between adjacent stops, we add extra stops at the end of the truck route with the same uniform and truncated normal distributions defined above. Take b (the completion time of the SMC solution for the original truck route) as an upper bound of the completion time for the PMCP. b is a valid upper bound because the postal service wants to keep the shift hours for the paired carriers similar to that of a single carrier. Then, we compute the number of extra stops in the following way. We add a stop each time, compute the completion time of the updated truck route using MIP4.1, and repeat the iteration if the completion time is less than b. We stop if the completion time of the updated truck route is greater than b. Thus, the number of extra stops is the maximum number of stops that we can add to the original truck route, while the completion time of the updated truck route for the PMCP is still below b, which ensures that the shift hours are not violated.

In the first two columns of Table 4.1, we give the values of D (the maximum travel time

D	μ	SMC	α	PMC	Savings (%)	Unproductive (%)	Runtime (s)	Extra Stops
2.5	2	64.59	2	50.58	21.73	30.64	0.35	5.09
			5	58.32	9.85	64.89	0.12	1.75
			10	61.41	5.03	81.20	0.06	0.67
	5	125.08	2	82.02	34.46	15.50	0.48	10.11
			5	92.65	22.78	24.62	0.21	5.63
			10	108.74	13.14	53.84	0.12	2.55
	10	224.51	2	131.94	41.24	8.75	0.50	13.74
			5	148.22	34.00	9.61	0.33	10.06
			10	170.61	24.05	22.61	0.20	6.38
5	2	90.63	2	80.53	11.19	52.23	0.23	2.22
			5	87.38	3.63	84.59	0.06	0.32
			10	89.24	1.55	91.46	0.05	0.10
	5	150.51	2	112.22	25.52	24.87	0.98	6.38
			5	132.40	12.15	56.36	0.33	2.55
			10	141.41	6.13	77.11	0.15	0.92
	10	251.26	2	163.19	35.08	15.64	0.42	10.36
			5	192.30	23.52	23.65	0.20	5.58
			10	217.53	13.48	51.34	0.13	2.74
10	2	140.15	2	134.54	4.04	76.67	0.11	0.58
			5	138.58	1.13	94.00	0.05	0.08
			10	139.37	0.56	96.81	0.02	0.06
	5	203.04	2	175.11	13.87	42.83	0.22	2.92
			5	193.80	4.68	78.70	0.07	0.54
			10	198.40	2.37	89.73	0.06	0.22
	10	299.56	2	222.42	25.85	24.72	0.32	6.74
			5	262.73	12.44	55.55	0.15	2.44
			10	281.00	6.37	76.69	0.06	0.85
Avg	-	-	-	-	15.18	52.76	0.20	3.30

Table 4.1: Results for MIP4.1 and the impact on a single truck. ("SMC": the average completion time over the 150 instances of the SMC solution. "PMC": the average completion time over the 150 instances of the PMC solution. "Savings": the average percentage savings in the completion time. "Unproductive": the average percentage of the time in the PMCP that mail carriers spend waiting at service stops and moving forward together using the truck. "Runtime": the average running time in seconds for MIP4.1 to find optimal solutions. "Extra Stops": the average number of extra stops added to the truck route using the savings in the completion time from the PMCP.)

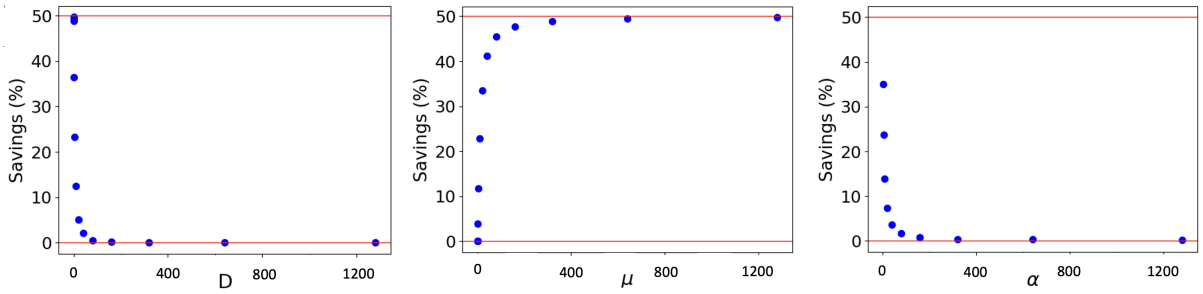
between adjacent stops) and the average service time "μ". For each combination of D and μ, we generate 150 random truck routes composed of 50 routes for $\sigma = 0.5,\ 1.0$, and 1.5. The average completion time over the 150 instances of the SMC solution is given in the column "SMC". Then, we test three different ratios of the truck speed to the walking speed "α". For each α, the average completion time over the 150 instances of the PMCP is reported in the column "PMC". The average percentage savings in the completion time are recorded in the column "Savings". The "Unproductive" column gives the average percentage of the time in the PMCP that mail carriers spend waiting at service stops and moving forward together using the truck. This column tells us in which configurations the extra mail carrier is most useful. For example, when $D = 2.5$, $\mu = 10$, and $\alpha = 2$, the unproductive percentage is very small, corresponding to large savings in the completion time. The "Runtime" column gives the average running time in seconds for MIP4.1 to find optimal solutions. The "Extra Stops" column gives the average number of extra stops added to the truck route using the savings in the completion time from the PMCP. In the last row of Table 4.1, we give the averages of the last four columns.

We observe that the percentage savings in the completion time of the PMCP vary from 0.56% to 41.24% with an average of 15.18%. For the most part, the percentage savings have a positive correlation with μ and a negative correlation with D and α. As μ increases, the idle leading mail carrier prefers to walk or driver forward to service stops. Both mail carriers are servicing simultaneously, so the PMCP has a large percentage savings in the completion time. When D or α increases, walking becomes less favorable compared to driving. Waiting at the service stops and moving forward using the truck together comprise a larger percentage of the completion time (see the column "Unproductive"). This leads to a smaller percentage savings in the completion time. A similar pattern can be observed in the column "Extra Stops", i.e., the

number of extra stops generated by the PMC setting increases as D decreases, α decreases, or μ increases. It is worth noting that with the completion time savings achieved by the PMCP, over 50% more stops can be serviced within the same shift hours. This could result in a significant capacity increase in practice if the SMC setting wants to achieve the same number of service stops. Moreover, the running times of MIP4.1 on these instances are small, with an average of 0.20 second. Recall that the calculation of the extra stop measure requires the MIP4.1 to be repeatedly solved with an increasing number of service stops. This calculation will not be possible if MIP4.1 is not so efficient.

To further characterize the relationship between the percentage savings in the completion time and the parameters D, μ, and α, we design the following computational experiment. Initially, the default values of these three parameters are set as $D = 5$, $\alpha = 5$, and $\mu = 10$. Then, we vary one parameter at a time, keep the other two parameters fixed as the control variables, and compute the percentage savings in the completion time with MIP4.1. When D or μ is chosen to vary, its domain is $\{1/32, 1/16, 1/8, 2, 5, 10, 20, 40, 80, 160, 320, 640, 1280\}$. Note that these values are chosen to find the asymptotic behavior of the savings in the completion time and hence might not be practical. When α is chosen to vary, because $\alpha > 1$ by assumption, the domain of α is $\{2, 5, 10, 20, 40, 80, 160, 320, 640, 1280\}$. For the test instances, we proceed in a similar fashion to the experiment in Table 4.1 and generate 150 random routes for each combination of D and μ. For example, when D is chosen to vary, 150 random truck routes are created for each value of $D \in \{1/32, 1/16, 1/8, 2, 5, 10, 20, 40, 80, 160, 320, 640, 1280\}$ with $\mu = 10$ and $\alpha = 5$ fixed. These sets of 150 routes are composed of 50 routes for each $\sigma = 0.5,\ 1$, and 1.5.

In Figure 4.6, we plot the relationships between the average percentage savings in the completion time and the parameters D, μ, and α (Recall that D is the maximum travel time



(a) Percentage savings vs. D. (b) Percentage savings vs. μ. (c) Percentage savings vs. α.

Figure 4.6: Asymptotic relationships between the percentage savings in the completion time and the parameters D, α, and μ. The red lines are the asymptotes. (D: the maximum travel time between adjacent stops. μ: the average service time. α: the ratio of the truck speed to the walking speed.)

between adjacent stops, μ is the average service time, and α is the ratio of the truck speed to the walking speed.). We find that the percentage savings converge to zero asymptotically in Figures 4.6a and 4.6c, as D or α becomes very large. Zero percentage savings indicates that the idle leading mail carrier will always choose to wait at each stop so that a second mail carrier is not needed. In Figure 4.6a, the percentage savings converge to 50% as D approaches zero. Recall that the completion time of the SMC solution is $L + \sum_{i=1}^{N} s_i$. With the PMCP, one mail carrier must spend at least $(\sum_{i=1}^{N} s_i)/2$ amount of time servicing stops. Each mail carrier spends at least L amount of time traversing the truck route. Hence, the percentage savings of the PMCP can never exceed 50%. Although 50% is an asymptotic limit, we can approach 50% with a realistic configuration, e.g., $D = 2.5, \mu = 10, \alpha = 2$ in Table 1. This configuration could correspond to postal deliveries in an urban area with congested traffic, where, for example, the average travel time between service stops is 2.5 minutes, the average service time of a stop is 20 minutes, the walking speed is 5 kilometers per hour, and the truck speed is 10 kilometers per hour. In Figure 4.6b, we observe that the percentage savings converge to 50% asymptotically as μ becomes very large. When μ is significantly larger compared to D, the mail carriers will evenly

split the service stops so the percentage savings are close to 50%. As μ approaches zero, waiting at the service stop becomes a better action and diminishes the benefits resulting from a second mail carrier. Based on these observations, we conclude that the PMC setting performs better in densely populated areas with traffic congestion and a long service time.

4.4.2 The Impact of the PMC Setting on a Fleet of Trucks

We have shown that the PMC setting on a single truck leads to completion time savings and extra stops. However, blindly adopting the PMC setting to all trucks might not be the most economical strategy. We must consider the cost of the second mail carrier ($27.32 per hour in 2019 [64]). To better understand the impact of the PMC setting on a fleet of trucks, we now quantify it from a cost analysis perspective, which finds the best trade-off between the cost of the second mail carrier and the savings from the capacity increase.

We consider a postal service that has a service area with many service stops. Servicing the area requires several trucks and carriers. In other words, we must manage a fleet of trucks. The travel time between adjacent stops and the service time for a stop are generated by the uniform distribution $\mathcal{U}(0, D)$ and the truncated normal distribution $\mathcal{N}(\mu, \sigma^2)$, as defined in Section 4.4.1. Given an upper bound on a mail carrier's shift hours, we are interested in the number of trucks required to service the area with an SMC fleet, a PMC fleet, and a hybrid fleet.

For an SMC or a PMC fleet, a truck (Truck 1) with one or two mail carriers begins from the start point of the service area, continues servicing the passing stops, and ends at the last stop, where the completion time of the current route is below the limit of the shift hours. In the case of a truck with paired carriers, the completion time of the route is computed by MIP4.1. For the

SMC and PMC fleets, if there still exist unserviced stops in the service area, then a second truck (Truck 2) is required, and the first unserviced stop in the area becomes the first stop of Truck 2. We assume that the travel time from the starting location of Truck 2 to its first service stop is governed by $\mathcal{U}(0, D)$. The set of stops that Truck 2 services is computed using the same method applied to Truck 1. We keep adding trucks until all stops in the service area are serviced. Because the carriers and trucks always service as many stops as possible in the SMC and PMC fleets, with respect to the limit of the shift hours, the numbers of trucks in the SMC and PMC fleets are minimized. Therefore, the costs in the SMC and PMC fleets are also minimized. We denote the number of trucks required in the SMC fleet by T_{SMC} and that in the PMC fleet by T_{PMC}.

For a hybrid fleet, we use a simple method to decide the mixture of the SMC trucks (i.e., trucks with a single carrier) and PMC trucks (i.e., trucks with paired carriers). Given the solution of a PMC fleet, we observe that the number of PMC trucks in the hybrid fleet could vary from 1 to $T_{PMC} - 1$. For each integer value from 1 to $T_{PMC} - 1$, we compute the number of trucks required by first using all PMC trucks, and then using the SMC trucks on the unserviced stops in the service area. Thus, the hybrid fleet obtained by our method always uses PMC trucks first and switches to SMC trucks later. The set of stops that a truck in the hybrid fleet services is computed using the same method applied in the SMC or PMC fleet, i.e., the truck and the carriers will service all passing stops until the completion time reaches the limit of the shift hours. We denote the number of trucks in the hybrid fleet by $T_{Hybrid} = T_{HSMC} + T_{HPMC}$, where T_{HSMC} and T_{HPMC} are the numbers of SMC and PMC trucks in the hybrid fleet, respectively.

In Figure 4.7, we use an example to illustrate how a hybrid fleet services the area. The start point of the trucks is located in the center and the service stops are scattered around the start point. To keep Figure 4.7 simple and readable, we only draw a small subset of 75 service stops.

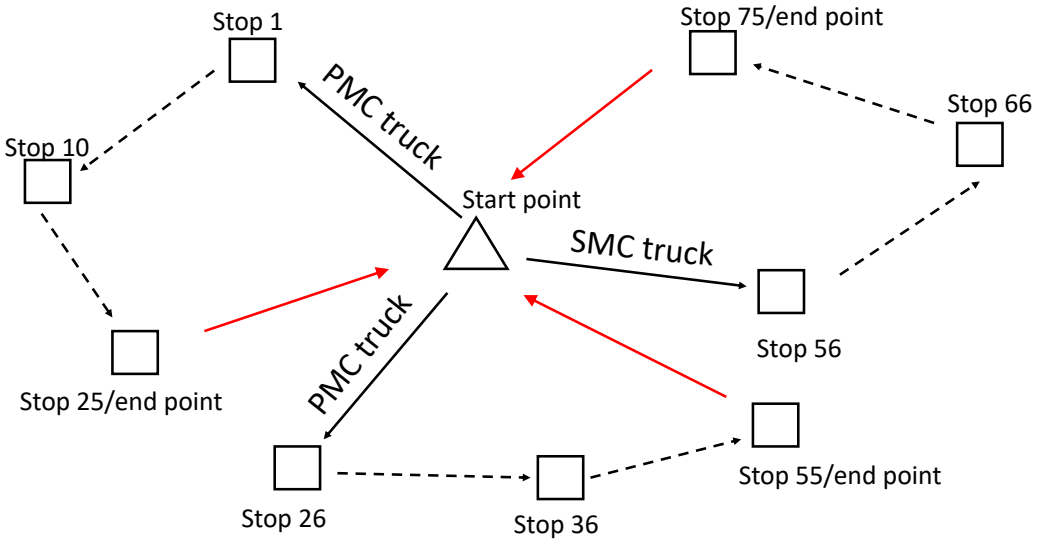


Figure 4.7: An example of a service area with 75 stops. Note that only a subset of stops are plotted here. Black solid arrow: a truck departs from the start point to service the first assigned stop. Dashed arrow connecting stops j and l: a truck and the carriers service stops $j, j+1, \ldots, l$. Red arrow: the truck and the carriers reach the limit of the shift hours after servicing the last stop (end point) and return to the start point.

A truck route is connected by a start point and an end point. The stop marked by the end point is the last stop so that the route completion time does not exceed the shift hours. We see that in the hybrid fleet, PMC trucks are deployed first and these trucks can visit more stops. Moreover, when a truck and the carriers reach the limit of the shift hours, a second truck is required and the first stop is the next unserviced stop in the area.

As expected, the PMC fleet and the hybrid fleet could reduce the number of trucks, compared to the SMC fleet, but they also require more mail carriers relative to the SMC fleet. Therefore, we must consider the costs associated with operating the three fleets. We use the information in Williams and Murray [64], which provides the operational costs of the trucking industry from 2011 to 2019. In particular, we are interested in the 2019 data including the marginal cost per hour of operating a truck ($37.79) and hiring a truck driver ($27.32) [64]. The truck-based costs

include fuel costs, lease or purchase payments, insurance premiums, repairs and maintenance, tolls, tires, and permits and licenses. The driver-based costs include wages and benefits. Using the information, the cost of an SMC fleet, a PMC fleet, and a hybrid fleet is $(37.79+27.32)T_{SMC}$, $37.79T_{PMC} + 2 \times 27.32T_{PMC}$, and $(37.79 + 27.32)T_{HSMC} + 37.79T_{HPMC} + 2 \times 27.32T_{HPMC}$, respectively. For a service area, we enumerate the costs of all $T_{PMC} - 1$ hybrid fleets and compare the lowest one with the SMC and PMC fleets. Among the SMC fleet, the PMC fleet, and the hybrid fleet with the lowest costs, we denote the number of trucks in the least-cost fleet by T_{LC} and the number of PMC trucks in the least-cost fleet by T_{LCPMC}. For example, if the least-cost fleet is the PMC fleet, then $T_{LC} = T_{LCPMC} = T_{PMC}$.

For our computational experiment, we set $\sigma = 1$ and the upper bound for a mail carrier's shift hours by using the column "SMC" in Table 4.1. For nine sets of values of $D,\ \mu$, and α, we record the averages of T_{SMC}, T_{PMC}, T_{LC}, and T_{LCPMC} among five randomly generated mail routes of 175, 200, and 225 stops in Table 4.2. The relative cost savings of a PMC or a hybrid fleet over an SMC fleet for a service area are given by $100((a - b)/a)\%$, where a is the cost of servicing the service area with the SMC fleet and b is that with the PMC or hybrid fleet. The column labeled as "RCS$_{PMC}$" (i.e., the relative cost savings for the PMC fleet) gives the average relative cost savings of the PMC fleets over the SMC fleets. The column labeled as "RCS$_{LC}$" (i.e., the relative cost savings for the least-cost fleet) gives that of the least-cost fleets over the SMC fleets. The last column labeled as "NOLCR" (i.e., the number of least-cost routes) reports the frequency in which the fleet has the lowest cost.

Over the 135 service areas, we observe that the SMC fleet has the lowest cost in 13 service areas, the PMC fleet has the lowest cost in 34 service areas, and the hybrid fleet has the lowest cost in 88 service areas. Thus, incorporating the PMC setting in a fleet of trucks leads to lower

D	μ	T_{SMC}	α	T_{PMC}	T_{LC}	T_{LCPMC}	RCS$_{PMC}$ (%)	RCS$_{LC}$ (%)	NOLCR
175 stops									
2.5	5	9.4	2	6.0	6.0	6.0	9.39	9.39	PMC(5)
			4	7.0	8.8	0.8	-5.72	2.81	SMC(3) Hybrid(2)
			5	7.0	9.0	0.6	-5.72	2.47	SMC(3) Hybrid(2)
2.5	10	9.6	2	6.0	6.0	5.0	11.27	15.65	Hybrid(5)
			4	6.0	6.0	5.6	11.27	13.02	PMC(3) Hybrid(2)
			5	6.2	6.2	5.6	8.32	10.94	PMC(4) Hybrid(1)
5	10	9.4	2	6.0	6.0	6.0	9.39	9.39	PMC(5)
			4	7.0	8.8	0.8	-5.72	2.81	SMC(3) Hybrid(2)
x			5	7.2	9.0	0.4	-8.74	2.45	SMC(3) Hybrid(2)
200 stops									
2.5	5	11.0	2	7.0	7.0	6.6	9.66	11.19	PMC(3) Hybrid(2)
			4	8.0	9.6	2.0	-3.24	5.10	Hybrid(5)
			5	8.0	9.8	2.0	-3.24	3.28	Hybrid(5)
2.5	10	11.0	2	6.0	6.0	6.0	22.57	22.57	PMC(5)
			4	7.0	7.0	6.0	9.66	13.48	Hybrid(5)
			5	7.0	7.0	6.8	9.66	10.42	PMC(4) Hybrid(1)
5	10	10.8	2	7.0	7.0	6.0	7.99	11.87	Hybrid(5)
			4	7.8	9.4	2.2	-2.53	4.42	Hybrid(5)
			5	8.0	10.0	0.8	-5.16	4.30	Hybrid(4) SMC(1)
225 stops									
2.5	5	12.0	2	8.0	8.0	6.6	5.36	10.26	Hybrid(5)
			4	9.0	10.2	3.4	-6.47	3.11	Hybrid(5)
			5	9.0	11.0	1.8	-6.47	2.04	Hybrid(5)
2.5	10	12.0	2	7.0	7.0	7.0	17.19	17.19	PMC(5)
			4	8.0	8.0	6.6	5.36	10.26	Hybrid(5)
			5	8.0	8.8	7.0	5.36	8.86	Hybrid(5)
5	10	12.0	2	8.0	8.0	6.8	5.36	9.56	Hybrid(5)
			4	9.0	10.6	2.4	-6.47	3.28	Hybrid(5)
			5	9.2	10.8	2.4	-8.84	1.61	Hybrid(5)

Table 4.2: Cost savings analysis for three types of fleets. Hybrid: a fleet of both SMC and PMC trucks. SMC: an SMC fleet. PMC: a PMC fleet. ("T_{SMC}": the number of trucks required in the SMC fleet. "T_{PMC}": the number of trucks required in the PMC fleet. "T_{LC}": the number of trucks in the least-cost fleet. "T_{LCPMC}": the number of PMC trucks in the least-cost fleet. "RCS$_{PMC}$": the relative cost savings for the PMC fleet. "RCS$_{LC}$": the relative cost savings for the least-cost fleet. "NOLCR": the number of least-cost routes.)

costs in 122 out of the 135 service areas (over 90%). The cost savings are 8.12% on average and can be as high as 22.57%. Furthermore, blindly using the PMC fleet does not always lead to the lowest cost. However, PMC trucks are most often used via a hybrid fleet. This highlights an important point: a hybrid configuration offers significant flexibility to deploy more PMC trucks in settings where they are cost-effective (small D, large μ, and small α). We see this flexibility result in the hybrid fleet being the least-cost fleet in 88 service areas versus 34 service areas for the PMC fleet.

We observe $T_{PMC} = T_{LC}$ with the least-cost fleets being all hybrid fleets for several cases in Table 4.2 (e.g., $D = 2.5, \mu = 10, \alpha = 2$, and 175 stops). This finding indicates that the service area can be best serviced by a combination of SMC and PMC trucks while keeping the total number of needed trucks the same as the PMC fleet. Compared to the PMC fleet, the hybrid fleet saves the cost of additional mail carriers. We notice that the values in columns RCS_{PMC} and RCS_{LC} are larger when D or α decreases and μ increases. This finding is consistent among the service areas with different stops. We also want to mention that the marginal cost per hour of operating a truck in Williams and Murray [64] does not include the carbon emission tax; hence, the cost savings are underestimated for both the PMC and hybrid fleets.

While we have shown that incorporating the PMC setting in a fleet results in significant cost savings, we want to emphasize that our results only underestimate the full impact of it. Our method to find a hybrid fleet is very simple and efficient, but cannot guarantee optimality. A method like dynamic programming can find the least-cost fleet. The trade-off is that the DP is more computationally expensive.

For the setup of the DP, we model the service area as a multigraph G. Since the truck can only move forward in the area, G is directed. Each node in G corresponds to a service stop. There

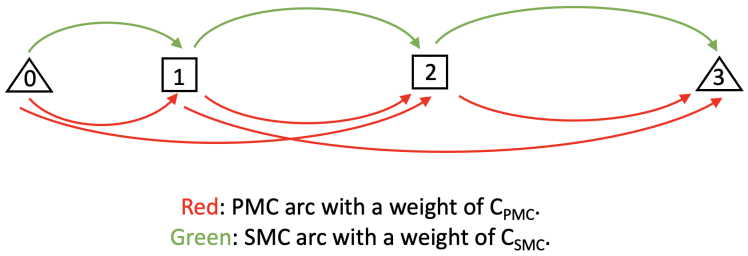


Figure 4.8: An example of G illustrating the SMC and PMC arcs.

are at most two arcs connecting a pair of nodes in G. These arcs are denoted by an SMC arc and a PMC arc. An SMC arc from nodes i to j exists if an SMC truck departs from the start point, services all stops between stops $i+1$ and j inclusive, and the completion time is within the limit of the shift hours. A PMC arc from nodes i to j exists if a PMC truck departs from the start point, services all stops between stops $i+1$ and j inclusive, and the completion time is within the limit of the shift hours. The weight of an SMC arc denoted by C_{SMC} is the costs of an SMC truck (a truck and a mail carrier), whereas the weight of a PMC arc denoted by C_{PMC} is the costs of a PMC truck (a truck and two mail carriers). Our objective of the DP is to find the shortest path connecting nodes 0 and $N+1$. In Figure 4.8, we plot an example of G, where we simply assume that an SMC truck can service a stop and a PMC truck can service two stops.

It is possible to have multiple SMC and PMC arcs originating from node i (see node 1 in Figure 4.8). Among all SMC (PMC) arcs originating from node i, suppose node j^i_{SMC} (j^i_{PMC}) is the last node that has an SMC (PMC) arc connected to node i (see Figure 4.9). In other words, for an SMC (PMC) truck servicing passing stops from stop $j+1$, stop j^i_{SMC} (j^i_{PMC}) is the last stop that the SMC (PMC) truck can service, such that the route completion time is within the limit of the shift hours.

Currently, there are many arcs in G. If an SMC truck services 20 stops on average and a

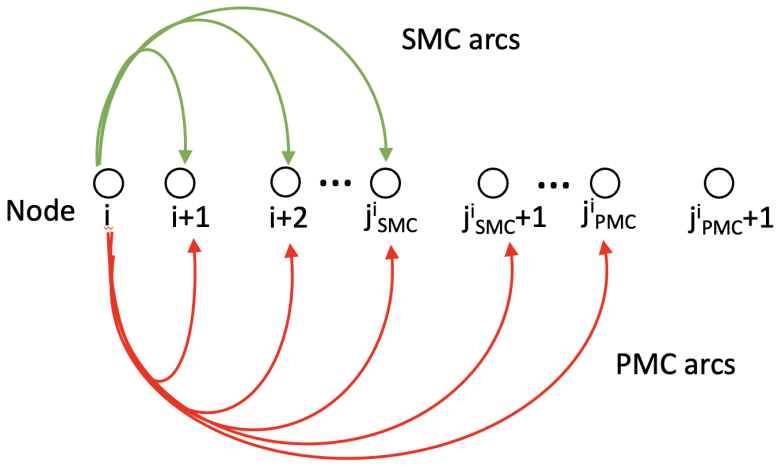


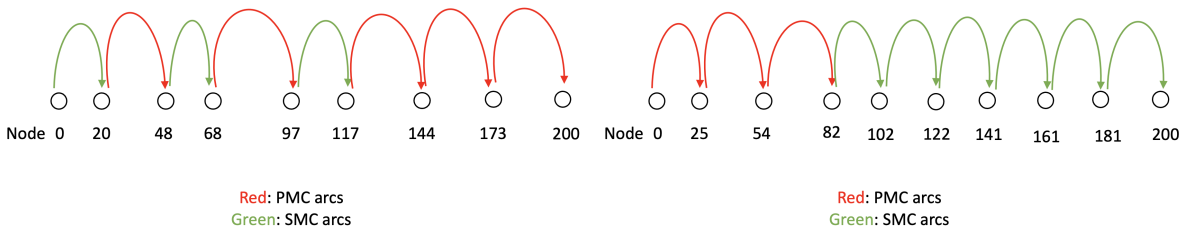
Figure 4.9: An example illustrating the SMC and PMC arcs originating from node i.

PMC truck services 25 stops on average, then there are around $45N$ arcs in G. We introduce an important proposition to reduce the search space and speed up the DP. We first denote the weight of the shortest path connecting stops i and $N+1$ by f_i. In the meantime, f_i also indicates the minimum cost for servicing all stops between stops i and $N+1$ inclusive. As i increases, f_i is non-increasing because fewer stops are required to be serviced. As a result, $f_i \leq f_j$ for any $j < i$. Now, we give the proposition.

Proposition 7 *It is never cost-effective for an SMC (PMC) truck to be used partially, i.e., for the truck to service all stops between stops $i+1$ and l inclusive, with $l < j^i_{SMC}$ (j^i_{PMC}).*

Proof: Included in Appendix C.

Proposition 7 is quite powerful since it dramatically reduces the search space of the DP to $2(N+1)$ arcs . In particular, for nodes $0 \leq i \leq N$, only arcs (i, j^i_{SMC}) and (i, j^i_{PMC}) are required in the DP. To solve the DP, we write down a recursive relationship for computing f_i by recognizing that, at each node, we make the decision of using an SMC or a PMC truck based on



(a) DP solution with a cost of $630.16 per hour.

(b) Simple method (PMC trucks first) solution with a cost of $640.63 per hour.

Figure 4.10: The hybrid fleets computed by two methods.

the minimum cost.

$$f_i = \min\left(f_{j^i_{SMC}} + C_{SMC}, f_{j^i_{PMC}} + C_{PMC}\right) \tag{4.14}$$

For the terminal condition of the DP, we set $f_{N+1} = 0$ and solve the remaining f_i using backward induction. The weight of the shortest path connecting nodes 0 and $N+1$ is associated with f_0.

In Figure 4.10, we compare the solution of the DP and the solution of our simple method with a 200-stop example using $D = 5, \mu = 10, \alpha = 4$. We observe a $10.47 reduction in the costs per hour using the DP. Furthermore, the DP has running times of 520 seconds versus our simple method has running times of 16 seconds. In conclusion, we believe our simple method and the DP give convincing results to show that the PMC setting can lead to positive impacts in the situation of a fleet of trucks.

4.4.3 Results of the Basic and Balanced Heuristics

In this section, we evaluate the performance of the two proposed heuristics. While MIP4.1 performs well on smaller instances, it can take a long time for larger instances. Thus, a good heuristic is invaluable in this situation.

We first demonstrate that the proposed heuristics can find a good-quality solution. We

N	μ	α	Basic Gap	Balanced Gap
10	5	2	8.05%	3.51%
	5	5	6.01%	2.65%
	10	2	7.20%	4.00%
	10	5	10.70%	4.49%
20	5	2	8.12%	3.75%
	5	5	6.01%	2.80%
	10	2	6.58%	4.15%
	10	5	11.26%	4.55%
30	5	2	7.94%	3.76%
	5	5	6.18%	3.06%
	10	2	6.85%	4.51%
	10	5	11.49%	4.88%
40	5	2	7.86%	3.62%
	5	5	5.93%	2.85%
	10	2	6.52%	4.51%
	10	5	11.29%	4.68%
Avg	-	-	8.00% (5.93% - 11.49%)	3.86% (2.65% - 4.88%)

Table 4.3: Optimality gap of the two proposed heuristics.

use the optimal solution generated by MIP4.1 to assess the performance of our heuristics. The optimality gap of a heuristic on a particular instance is defined as $100((c-b)/b)\%$, where b is the route completion time of the MIP4.1 solution and c is that of the heuristic solution. For the computational experiment, we fix $D = 5$ and vary μ and α because D and α have the same correlation to the completion time of the PMCP, so varying one of them is sufficient. For each combination of N and μ, the optimality gap is averaged over 150 instances, composed of 50 instances for $\sigma = 0.5,\ 1.0$, and 1.5. We show the results in Table 4.3. The "Basic Gap" column gives the average optimality gap for the basic heuristic. The "Balanced Gap" column gives that for the balanced heuristic. The last row of Table 4.3 records the averages and ranges of the columns "Basic Gap" and "Balanced Gap". We see that the basic heuristic always performs worse than the balanced heuristic. The optimality gap of the balanced heuristic ranges from

2.65% to 4.88%, with an average of 3.86%. Both heuristics are very fast with running times on the order of milliseconds for $N = 40$. The parameters α and μ appear to have no noticeable impact on the runtimes of both heuristics. We point out that the optimality gaps of our heuristics appears to be similar across various sized instances and believe this is attributable to the structure of the heuristics that make sequential decisions using a consistent rule throughout the route.

N	μ	α	Basic Gap	Balanced Gap	Basic Savings	Balanced Savings
50	5	2	7.94%	4.11%	19.84%	22.86%
	5	5	6.17%	2.70%	7.28%	10.16%
	10	2	6.53%	4.45%	30.81%	32.09%
	10	5	11.10%	4.24%	14.83%	20.09%
60	5	2	7.68%	3.57%	19.85%	22.88%
	5	5	6.61%	2.85%	7.13%	9.94%
	10	2	6.49%	4.47%	30.77%	31.94%
	10	5	10.98%	4.52%	14.65%	19.76%
70	5	2	7.86%	3.74%	19.65%	22.67%
	5	5	5.96%	2.47%	7.07%	9.86%
	10	2	6.78%	4.97%	30.73%	31.90%
	10	5	11.34%	4.11%	14.68%	19.87%
80	5	2	8.23%	3.75%	19.71%	22.82%
	5	5	5.98%	3.00%	6.96%	9.81%
	10	2	6.24%	4.42%	30.82%	31.97%
	10	5	12.21%	4.54%	14.79%	20.08%
90	5	2	-	-	19.91%	22.88%
	5	5	-	-	7.09%	9.98%
	10	2	-	-	30.97%	32.14%
	10	5	-	-	14.93%	20.16%
100	5	2	-	-	19.67%	22.66%
	5	5	-	-	6.87%	9.65%
	10	2	-	-	30.95%	32.07%
	10	5	-	-	15.11%	20.28%
Avg	-	-	8.01%	3.87%	18.13%	21.19%

Table 4.4: Optimality gap and percentage savings in the route completion time of the two proposed heuristics.

When $N > 50$, MIP4.1 takes more time to find the optimal solution. For $50 \leq N \leq 80$,

it often takes around 30 minutes to solve the MIP4.1 optimally. For $N > 80$, MIP4.1 struggles to find the optimal solution within two hours. To assess the performance of both heuristics, for $50 \leq N \leq 80$, we compute the optimality gap among 10 instances of $\sigma = 1$ for each combination of N and μ. The runtime limit of MIP4.1 is set to 30 minutes. If the optimal solution is not found within 30 minutes, we use the lower bound found by Gurobi to figure out the optimality gap. In addition, for $50 \leq N \leq 100$, we compute the percentage savings in the completion time of the heuristic solutions over the SMC solutions for all 150 instances.

We present the results in Table 4.4. The "Basic Gap" and "Balanced Gap" columns give the optimality gaps of the heuristics. The "Basic Savings" column gives the average percentage savings for the basic heuristic. The "Balanced Savings" column gives that for the balanced heuristic. The last row of Table 4.4 records the averages. The performance of both heuristics does not deteriorate when N increases. For the optimality gap, the results are almost identical to Table 4.3. For the percentage savings, when we fix μ and α and vary N, the percentage savings are very similar. The percentage savings in Table 4.4 show that the PMC setting is effective when $N \geq 50$, with an average savings of 21.19% given the fact that the percentage savings of the heuristics are a lower bound of the potential percentage savings. Both heuristics are still fast, with running times on the order of milliseconds for $N = 100$.

4.5 Extension to k Mail Carriers Per Truck

So far, we have been focusing on the case with two mail carriers per truck given its practicality. However, it is interesting to examine the case with three or more mail carriers per truck. In this section, we briefly discuss the general case in which there are k mail carriers per truck,

where k can be any positive integer number. We show that the basic heuristic can be extended for this general case and present some computational results to demonstrate that the marginal savings decrease as k increases. We also present an MIP formulation for the general case in Appendix E.

4.5.1 Extending the Basic Heuristic for the General Case

For k mail carriers per truck, we extend the basic heuristic and define a more general rule, i.e., the truck is always left with the last mail carrier and the remaining mail carriers choose to walk or wait. In particular, for an idle mail carrier A, we define $currStop$ as the current location of A and $nextStop$ as the first stop that has not been serviced or assigned to any mail carrier. Let TW be the set of mail carriers that are trailing A and $RT(tw)$ be the remaining service time of a mail carrier $tw \in TW$. Then, we compute

- $walkTime = \alpha(d_{nextStop} - d_{currStop})$,
- $waitTime = max_{tw \in TW}(d_{nextStop} - d_{currStop} + RT(tw))$.

If $walkTime < waitTime$, then mail carrier A should walk to $nextStop$ and service $nextStop$. If $waitTime \leq walkTime$, then the mail carrier should wait for the truck and ride with it until arriving at $nextStop$. Then, a mail carrier is randomly selected to service $nextStop$.

For the computational experiments, we fix $D = 5$. We vary the parameters $N = 10, 50, 100$, $\alpha = 2, 5, 10$, $\mu = 2, 5, 10$, $\sigma = 0.5, 1, 1.5$, and $k = 2, 3, 4, 5$. For each value of k, we compute the percentage savings in the completion time of k mail carriers per truck over an SMC per truck. The percentage savings are averaged over 4,050 instances composed of 50 instances for each combination of N, α, μ, and σ. The running times of the heuristic are very fast, with an average on the order of centiseconds.

k	Average Savings (%)	Marginal Savings (%)
2	10.24	10.24
3	12.43	2.19
4	13.47	1.05
5	14.13	0.66

Table 4.5: Percentage savings and marginal percentage savings in the completion time for $k \geq 2$ mail carriers per truck over the SMC setting.

We present the results in Table 4.5, where the "Average Savings" column gives the average percentage savings in the completion time of k mail carriers per truck over an SMC per truck. The "Marginal Savings"'s column gives the difference in the average percentage savings of k mail carriers per truck over $k-1$ mail carriers per truck. We observe that as k increases, the average percentage savings increase. However, the marginal percentage savings are decreasing, i.e., it is not effective to add an extra mail carrier in the truck as k increases. In fact, when $k = 2$, the largest value of the marginal percentage savings is generated, which is the PMC option.

4.6 Conclusions

The Paired Mail Carrier Problem may be useful to postal services that seek to reduce the size of their truck fleets or that wish to speed up the route completion time of an existing truck route. We developed an MIP formulation and two fast heuristics for the PMCP. Using the exact solution generated by the MIP formulation, we evaluate the impact of the PMC setting on both a one-truck situation and a fleet (multiple trucks) situation, relative to the SMC setting. On the one hand, considering one truck, the PMC setting can accomplish over 50% extra work within the same shift hours. On the other hand, considering a fleet of trucks, the PMC setting can lead to 22% cost savings. Overall, we demonstrate that the PMC setting as a novel operation mode

is worth considering in practice. We believe that using the PMC setting is most appropriate in areas with a high density of stops, relatively slow traffic, and long service times per stop. This combination of factors may occur in some urban environments, where apartment buildings or businesses are tightly packed together. To reduce pollution and traffic congestion, urban areas may be the most inclined to reduce the number of trucks on the road in the first place. Both heuristics were fast: the largest instances were solved on the order of milliseconds. The balanced heuristic produced accurate results, with an optimality gap of less than 5% for all instances. We presented an MIP formulation and an extension of the basic heuristic for the general case with three or more mail carriers per truck. We found that when there are two mail carriers per truck, the marginal percentage savings in the completion time were the largest. Future research could include an investigation of more sophisticated methods to form a hybrid fleet. Moreover, we are interested in balancing the workload among mail carriers. In our current model, if there are multiple mail carriers in the truck arriving at a stop, we randomly pick one to service the stop. In some situations, one mail carrier may service all stops, while other mail carriers just wait in the truck. We could correct this by scoring the difficulty of driving, walking, and servicing. Then, we could use a greedy algorithm to assign the more difficult tasks to the idle mail carriers who have lower difficulty scores at the time. There is also the potential to develop more sophisticated heuristics that consider a larger search space and additional improvement operators. Lastly, designing efficient algorithms for the DP formulation is another interesting research direction.

Chapter 5: Conclusions and Future Work

In this dissertation, we developed solution approaches for solving three specific applications of vehicle routing problems. First, we discussed the application of police patrol to a single vehicle. For a region with a high-crime neighborhood, we designed an accurate metric to measure the distance between the tour and the high-crime neighborhood. We proposed an energy function yielding a smooth transition from the L-dominated phase to the C-dominated phase. We introduced triangular paths, such that the patrol routes are sensible and close to the high-crime neighborhood. We applied the IP to show that in both a Euclidean graph and a grid network and under the same constraints on the travel length, the triangular path tour has a shorter distance to the center than the direct path tour.

Second, for a fleet of vehicles patrolling a region with a high-crime neighborhood, we focused on our attention on requiring at least one vehicle in the high-crime neighborhood at any given time during the patrol. We developed two solution approaches (global and partition) to find the patrol routes. In the global approach, we designed an IP formulation to find the optimal routes in a single run. In the partition approach, we followed a divide-and-conquer strategy. We first partitioned the region geographically and solved the routing problem in each subregion independently using the IP. The partition approach serves as a heuristic and can find high-quality solutions quickly. Column generation was also applied to improve the partition solution. To

determine the starting location of the routes, we presented a simple and efficient search algorithm.

Third, we considered the application of mail deliveries. By allowing two carriers per truck, we sought to reduce the size of truck fleets or speed up the completion time of an existing truck route. We developed an MIP formulation and two fast heuristics for the PMCP. Using the exact solution generated by the MIP formulation, we evaluated the impact of the PMC setting on both a one-truck situation and a fleet (multiple trucks) situation relative to the SMC setting. We showed that the PMC setting can lead to great savings in completion time and fleet size, especially in areas with a high density of stops, relatively slow traffic, and long service times per stop. This combination of factors may occur in some urban environments, where apartment buildings or businesses are tightly packed together.

5.1 Future Work

For the police patrol application, our current setup is deterministic, i.e., the patrol routes do not vary on a daily basis and may become predictable to offenders. We plan to introduce randomness into our model, such as increasing the prizes associated with uncovered locations after each patrol and re-optimizing routes constantly.

For the mail delivery application, our current model does not distinguish between mail carriers. If there are multiple mail carriers in the truck arriving at a stop, we randomly pick one to service the stop. In some situations, one mail carrier may service all stops, while other mail carriers just wait in the truck. We plan to score the difficulty of driving, walking, and servicing and assign difficult tasks to a mail carrier with a lower difficulty score. In addition, as autonomous driving technology emerges, we plan to analyze the impact on the completion time

and costs when a mail carrier with an autonomous vehicle is applied.

Appendix A: Technical Proofs In the HSCPP

Proposition 1 *The HSCPP is NP-hard.*

Proof: We reduce the team orienteering problem (TOP) [13] to the HSCPP. In the TOP, each location is associated with a prize and a travel time limit is fixed for each vehicle. Given a start and end point, the TOP involves finding a set of paths from the start point to the end point to maximize the total prize. Consider a special instance of the HSCPP in which there are only K nodes in the HCN and these K nodes share the same location. Then the optimal solution of this special case solves the TOP, assuming that the start and end point are one of the K nodes in the HCN. Therefore, the HSCPP is at least as difficult as the TOP and hence is NP-hard.

Proposition 2 *Given a pair of nodes $i, j \in V$ that satisfies $a_{ir} = a_{jr},\ \forall r \in \mathcal{R}'$, then there exists a dual-optimal solution satisfying $\alpha_i^* = \alpha_j^*$.*

Proof: We first present the dual formulation of LP4R:

$$\text{Dual of LP4R}: \quad \min \qquad \textstyle\sum_{l\in V} \alpha_l - T\beta + K\gamma \tag{A.1}$$

$$\text{subject to} \quad \textstyle\sum_{l\in V} a_{ir}\alpha_l - IT_r\beta + \gamma \geq c_r, \quad \forall\, r \in \mathcal{R}', \tag{A.2}$$

$$\alpha_l \geq 0, \quad \forall\, l \in V, \tag{A.3}$$

$$\beta, \gamma \geq 0. \tag{A.4}$$

Let $(\bar{\boldsymbol{\alpha}}, \bar{\beta}, \bar{\gamma})$ be an optimal solution for the dual of LP4R. We construct a solution for the dual variable $\boldsymbol{\alpha}$ such that

$$\alpha_l^* = \begin{cases} \frac{\bar{\alpha}_i + \bar{\alpha}_j}{2}, & l \in \{i, j \in V : i \neq j \text{ and } a_{ir} = a_{jr},\ \forall r \in \mathcal{R}'\}, \\ \bar{\alpha}_l, & \text{otherwise.} \end{cases} \tag{A.5}$$

Because $\sum_l \alpha_l^* = \sum_l \bar{\alpha}_l$, we show that the dual solution ($\boldsymbol{\alpha}^*$, $\bar{\beta}$, $\bar{\gamma}$) is feasible, and hence optimal. Given that $a_{ir} = a_{jr},\ \forall r \in \mathcal{R}'$, we know that a tour either includes both nodes i and j or neither of them. Given a tour $r \in \mathcal{R}'$ that includes both nodes i and j, the associated dual constraint is

$$\bar{\alpha}_i + \bar{\alpha}_j + \sum_{l \in V \setminus \{i,j\}} a_{lr} \bar{\alpha}_l - IT_r \bar{\beta} + \bar{\gamma} \geq c_r. \tag{A.6}$$

Now, we use (A.5) to replace $\bar{\alpha}$ in (A.6) by α^*. We obtain

$$\alpha_i^* + \bar{\alpha}_j^* + \sum_{l \in V \setminus \{i,j\}} a_{lr} \alpha_l^* - IT_r \bar{\beta} + \bar{\gamma} = \bar{\alpha}_i + \bar{\alpha}_j + \sum_{l \in V \setminus \{i,j\}} a_{lr} \bar{\alpha}_l - IT_r \bar{\beta} + \bar{\gamma} \geq c_r. \tag{A.7}$$

Given a tour $r \in \mathcal{R}'$ that includes neither nodes i nor j, the associated dual constraint is

$$\sum_{l \in V \setminus \{i,j\}} a_{lr} \bar{\alpha}_l - IT_r \bar{\beta} + \bar{\gamma} \geq c_r. \tag{A.8}$$

Replacing $\bar{\alpha}$ by α^* will not change (A.8). Overall, we conclude that ($\boldsymbol{\alpha}^*$, $\bar{\beta}$, $\bar{\gamma}$) is feasible. Because it is also dual-optimal and satisfies the property that $\alpha_i^* = \alpha_j^*$, the proof is completed.

Claim 1 *For $1 \leq k \leq K$, the starting point of tour k is in Outer$_k$.*

Proof: We prove this claim by contradiction. Assume that the starting point of the tour is located

in Inner$_k$. Then, the coverage of inner nodes is broken into two time segments separated by visits to outer nodes. This contradicts the operation requirement saying that a tour must visit all inner nodes in a continuous time window.

Appendix B: IP Formulation of the ESPPRC Pricing Model

Let $x_{ij} = 1$ if arc (i, j) is included in the shortest path; otherwise $x_{ij} = 0$. We have the following IP formulation for the ESPPRC using inner node i^* as the source and sink.

$$\text{ESPPRC}_{i^*}: \quad \min \quad \sum_{i \in V_{i^*}} \sum_{j \in V_{i^*}} w_{ij} x_{ij} \tag{B.1}$$

$$\text{subject to} \quad \sum_{i \in V_{i^*}} \sum_{j \in V_{i^*}} x_{ij} - \sum_{i \in V_{i^*}} \sum_{j \in V_{i^*}} x_{ji} = \begin{cases} 1, & \text{for } i = s, \\ -1, & \text{for } i = t, \\ 0, & \text{otherwise.} \end{cases} \tag{B.2}$$

$$\sum_{j \in V_{i^*}} x_{ij} \leq 1, \quad \forall i \in V_{i^*}, \tag{B.3}$$

$$\sum_{i \in V_{i^*}} \sum_{j \in V_{i^*}} L_{ij} x_{ij} \leq \tau, \tag{B.4}$$

$$\sum_{i \in V_{\text{inner},i^*}} \sum_{j \in V_{\text{outer}}} x_{ij} = 1, \tag{B.5}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq \sum_{i \in S \setminus k} \sum_{j \in V} x_{ij}, \quad \forall S \subset V_{i^*} \setminus \{s, t\}, k \in S, \tag{B.6}$$

$$x_{ij} \in \{0, 1\}, \quad \forall\, i, j \in V_{i^*}. \tag{B.7}$$

The objective function (B.1) minimizes the weight of the shortest path. Constraints (B.2) force there to be an outgoing arc from s and an incoming arc to t. They also balance the flow at other nodes. Constraints (B.3) ensure that every node is visited at most once. Constraint (B.4) limits the length of the shortest path. Constraint (B.5) ensures that the shortest path only enters the

HCN once. Constraints (B.6) are the subtour elimination constraints. Constraints (B.7) give the domain of the variables.

Appendix C: Proofs of the Propositions In the PMCP

Proposition 3 *An idle mail carrier should never wait partially.*

Proof: It is never better for an idle mail carrier to wait partially. An idle mail carrier A should either move forward to the next stop immediately (by walking or driving) or wait the full duration until the trailing mail carrier has returned to the truck so both can ride together. If A waits partially, we can always replace the partial wait with an action, resulting in a solution with a better (or not worse) completion time.

Assume that A waits partially for θ minutes at stop j. We consider the action taken by A after the partial wait. In the case that A is the leading mail carrier at stop j. If A moves forward by driving alone, we replace the partial wait by driving forward immediately. This reduces the arrival time of A at stop $j+1$ by θ minutes and leads to a better (or not worse) completion time of the PMCP. If A moves forward by walking and mail carrier B catches up to A using the truck before stop $j+1$, we replace the partial wait with a full wait at stop j. The arrival time of A at stop $j+1$ using the full wait is the same as that using the partial wait; hence, the completion time of the PMCP is not worse. If A moves forward by walking and mail carrier B does not catch up to A before stop $j+1$, we replace the partial wait by walking forward immediately. Carrier A can arrive at stop $j+1$ θ minutes earlier and the completion time of the PMCP is better (or not worse). In the case that A is the trailing mail carrier, a full wait is not possible. Any partial wait

only delays the completion time of A. Thus, an immediate action of A leads to a better (or not worse) completion time of the PMCP.

For all these cases, by taking an immediate action, the completion time of the PMCP will never get worse and might be improved by θ minutes. Therefore, taking an immediate action dominates a partial wait.

Proposition 4 *It is never advantageous for a leading mail carrier to walk partially, i.e., for the leading mail carrier to walk only part of the way from stops j to $j+1$, then wait to be picked up by the trailing one in the truck.*

Proof: If the leading mail carrier walks partially between stops j and $j+1$, both mail carriers reach stop $j+1$ at the same time in the truck. However, the same arrival time for both carriers and the truck at stop $j+1$ can be achieved with the leading mail carrier waiting for the truck at stop j. We prefer the latter solution because it does not involve walking and is better in terms of saving the leading mail carrier's physical energy.

Proposition 5 *There exists an optimal solution such that if a walking mail carrier gets into a non-empty truck (i.e., the other mail carrier is in the truck.), it happens at a stop.*

Proof: Assume that this proposition is not true. Then, we have an optimal solution where a walking mail carrier gets in a non-empty truck at a point between stops j and $j+1$. There are two cases. First, the leading mail carrier drives the truck to some location between stops j and $j+1$ and waits for the trailing one to catch up on foot. We denote the leading mail carrier's waiting location by $d_j + \beta(d_{j+1} - d_j)$, where $0 < \beta < 1$. After the trailing mail carrier catches up and gets in the truck, both carriers arrive at stop $j+1$ at the same time. Let $D_{t,j}$ be the

departure time of the trailing mail carrier at stop j. Then, both carriers reach stop $j+1$ at time $D_{t,j} + \alpha\beta(d_{j+1} - d_j) + (1-\beta)(d_{j+1} - d_j) = D_{t,j} + (\alpha\beta + 1 - \beta)(d_{j+1} - d_j)$. However, a better solution can be constructed in the following way. Let the leading mail carrier wait at stop j. Then, both mail carriers leave stop j at time $D_{t,j}$ and reach stop $j+1$ at time $D_{t,j} + (d_{j+1} - d_j)$. Given that $\alpha > 1$ and all other parts of the solution stay the same as before, we reduce the completion time of the PMCP by $(\alpha - 1)\beta(d_{j+1} - d_j)$. A contradiction is found. The other situation is that the leading mail carrier walks to some location between stops j and $j+1$ and waits to be picked up by the truck driven by the trailing one. This is the partial walk in Proposition 4. As a result, a contradiction can be found for both cases.

Proposition 6 *In a solution of the PMCP, the truck is always between mail carriers, i.e.,* $\min(loc(A,t), loc(B,t)) \leq loc(\mathcal{T},t) \leq \max(loc(A,t), loc(B,t))$ *for any* t.

Proof: Recall that the truck and the carriers can only move forward on the truck route. Moreover, for the truck to move forward, one of the mail carriers must be in the truck. If the truck is behind two mail carriers, one mail carrier must walk backwards to retrieve the truck so that the truck can reach the end of the route. This cannot happen because the mail carriers do not walk backwards. On the other hand, for the truck to be in front of two mail carriers, one mail carrier parks the truck and walks backwards on the truck route. This also cannot happen for the same reason as above.

Proposition 7 *It is never cost-effective for an SMC (PMC) truck to be used partially, i.e., for the truck to service all stops between stops* $i+1$ *and* l *inclusive, with* $l < j^i_{SMC}$ (j^i_{PMC}).

Proof: Without loss of generality, we assume that an SMC truck is used partially. Then, the truck ends the servicing at stop l and the minimum costs of servicing the remaining area is

f_{l+1}. However, if the truck ends the servicing at stop j^i_{SMC}, the minimum costs of servicing the reaming area is $f_{j^i_{SMC}+1}$. Since $f_{j^i_{SMC}+1} \leq f_{l+1}$, it is always better for the truck to service as many stops as possible subject to the limit of the shift hours.

Appendix D: DP Formulation for the PMCP

In this section, we model the PMCP as a dynamic programming (DP) model. We first show the notation of the DP. Let $F(loc_{ld}, s_{ld}, loc_{tr}, s_{tr}, loc_T)$ be the minimum completion time from the following state: loc_{ld} is the location on the truck route of the leading mail carrier. s_{ld} is the service time at loc_{ld} of the leading carrier. loc_{tr} is the location on the truck route of the trailing mail carrier. s_{tr} is the service time at loc_{tr} of the trailing carrier. loc_T is the location on the truck route of the truck. In addition, the terminal condition of the DP is $F(d_N, 0, d_N, 0, d_N) = 0$. Thus, $F(0, 0, 0, 0, 0)$ gives the minimum completion time of the PMCP. For the recursive relationship in the DP, we first show that every state of the DP can be reduced to the state $(loc_{ld}, 0, loc_{tr}, s_{tr}, loc_T)$ (i.e., the leading mail carrier is idle) and then give a recursive formula for the state $(loc_{ld}, 0, loc_{tr}, s_{tr}, loc_T)$.

Let $s_{ld} > 0$. Given a state $(loc_{ld}, s_{ld}, loc_{tr}, s_{tr}, loc_T)$, we let both carriers service for $\min(s_{ld}, s_{tr})$ minutes and reach the following relationship.

$$F(loc_{ld}, s_{ld}, loc_{tr}, s_{tr}, loc_T) = \begin{cases} s_{ld} + F(loc_{ld}, 0, loc_{tr}, s_{tr} - s_{ld}, loc_T), & s_{ld} \leq s_{tr}, \\ \\ s_{tr} + F(loc_{ld}, s_{ld} - s_{tr}, loc_{tr}, 0, loc_T), & \text{otherwise.} \end{cases} \tag{D.1}$$

In the first case, the reduction is complete. In the second case, it indicates that the trailing carrier is free to move forward to catch the leading one. Recall that the truck is always in between of the two mail carriers. We have three subcases. First, the trailing mail carrier does not reach the truck after $s_{ld} - s_{tr}$ minutes. Second, the trailing mail carrier reaches the truck but does not reach loc_{ld} after $s_{ld} - s_{tr}$ minutes. Third, the trailing mail carrier reaches loc_{ld} within $s_{ld} - s_{tr}$ minutes and becomes the leading mail carrier. Then,

$$
\begin{aligned}
&F(loc_{ld}, s_{ld} - s_{tr}, loc_{tr}, 0, loc_T) = \\
&\begin{cases}
s_{ld} - s_{tr} + F(loc_{ld}, 0, loc_{tr} + (s_{ld} - s_{tr})/\alpha, 0, loc_T), & \text{if } s_{ld} - s_{tr} \leq \alpha(loc_T - loc_{tr}), \\
s_{ld} - s_{tr} + F(loc_{ld}, 0, & \text{if } \alpha(loc_T - loc_{tr}) < s_{ld} - s_{tr} \leq \\
loc_T + (s_{ld} - s_{tr} - \alpha(loc_T - loc_{tr})), 0, & \alpha(loc_T - loc_{tr}) + loc_{ld} - loc_T, \\
loc_T + (s_{ld} - s_{tr} - \alpha(loc_T - loc_{tr}))), & \\
(\alpha(loc_T - loc_{tr}) + loc_{ld} - loc_T) & \text{otherwise.} \\
+F(loc_{ld}, 0, loc_{ld}, s_{ld} - s_{tr} - & \\
(\alpha(loc_T - loc_{tr}) + loc_{ld} - loc_T), loc_{ld}), &
\end{cases}
\end{aligned}
\tag{D.2}
$$

The reduction is complete for all three subcases because they are all in a state where the leading mail carrier is idle. Thus, we can always reach a state $(loc_{ld}, 0, loc_{tr}, s_{tr}, loc_T)$ where the leading mail carrier is idle.

The remaining task is to derive a recursive formula for the state $(loc_{ld}, 0, loc_{tr}, s_{tr}, loc_T)$. Based on our propositions presented in Section 4.2, the idle leading carrier can wait for the trailing mail carrier or move forward alone by driving or walking. Given that the next unserviced stop is j, we use β to denote the portion of travel from loc_{ld} to stop j done by the leading carrier using the truck. $T(\beta)$ is the travel time of the leading mail carrier from loc_{ld} to stop j, which equals to $\beta(d_j - loc_{ld}) + \alpha(1-\beta)(d_j - loc_{ld})$. The recursive formula for $F(loc_{ld}, 0, loc_{tr}, s_{tr}, loc_T)$ is

$$F(loc_{ld}, 0, loc_{tr}, s_{tr}, loc_T) = \min\{s_{tr} + \alpha(loc_T - loc_{tr}) + (d_i - loc_T) + F(d_j, 0, d_j, s_j, d_j), \\ \min_{\beta}[T(\beta) + F(d_j, s_{ld}(\beta), loc_{tr}(\beta), s_{tr}(\beta), loc_T(\beta))]\}, \tag{D.3}$$

where $loc_{tr}(\beta)$, $s_{ld}(\beta)$, $s_{tr}(\beta)$, and $loc_T(\beta)$ are all functions of β. The first term in the minimization corresponds with the leading mail carrier waiting for the trailing one. The second term in the minimization corresponds with the leading mail carrier moving forward alone.

If $loc_{ld} = loc_T$, i.e., the truck is with the leading mail carrier, then β can take any value between 0 and 1 in (D.3). Depending on the size of $T(\beta)$, there are four cases for the location of the trailing mail carrier $loc_{tr}(\beta)$. First, the trailing mail carrier does not move forward if $s_{tr} \geq T(\beta)$ minutes. Second, the trailing mail carrier finishes servicing but does not reach the truck after $T(\beta)$ minutes. Third, the trailing mail carrier reaches the truck but does not reach stop j after $T(\beta)$ minutes. Lastly, the trailing mail carrier catches the leading one within $T(\beta)$

minutes. The values of $loc_{tr}(\beta)$ for all four cases are given below.

$$loc_{tr}(\beta) = \begin{cases} loc_{tr}, & \text{if } s_{tr} \geq T(\beta), \\ loc_{tr} + (T(\beta) - s_{tr})/\alpha, & \text{if } s_{tr} < T(\beta) \leq s_{tr} + \\ & \alpha\beta(d_j - loc_{ld}) + \alpha(loc_T - loc_{tr}), \\ loc_{ld} + \beta(d_j - loc_{ld}) + (T(\beta) & \text{if } s_{tr} + \alpha\beta(d_j - loc_{tr}) + \alpha(loc_T - loc_{tr}) \\ -(s_{tr} + \alpha\beta(d_j - loc_{tr}) & < T(\beta) \leq s_{tr} + \alpha\beta(d_j - loc_{ld}) + \\ +\alpha(loc_T - loc_{tr}))), & \alpha(loc_T - loc_{tr}) + (1 - \beta)(d_j - loc_{ld}), \\ d_j, & \text{otherwise.} \end{cases} \tag{D.4}$$

For the location of the truck, the truck is at the parking location if the trailing mail carrier has not caught the truck. Otherwise, the truck is with the trailing mail carrier. As a result, $loc_T(\beta) = \max\{loc_{ld} + \beta(d_j - loc_{ld}), loc_{tr}(\beta)\}$.

For the service time of the trailing carrier, we have three cases. First, the service time can be the difference between s_{tr} and $T(\beta)$ if $s_{tr} \geq T(\beta)$. Second, if the trailing mail carrier catches the leading one before stop j, then two mail carriers arrive at stop j together in the truck. The service time of the trailing mail carrier is s_j because the trailing one services stop j and the leading one is ready for the next action. Otherwise, the service time of the trailing carrier is zero.

This implies

$$s_{tr}(\beta) = \begin{cases} s_{tr} - T(\beta), & \text{if } s_{tr} \geq T(\beta), \\ s_j, & \text{if } T(\beta) \geq s_{tr} + \alpha\beta(d_j - loc_{ld}) + \alpha(loc_T - loc_{tr}) + (1-\beta)(d_j - loc_{ld}), \\ 0, & \text{otherwise.} \end{cases} \tag{D.5}$$

For the service time of the leading mail carrier, we have two cases. If the trailing carrier catches the leading one, then the leading mail carrier has service time 0. Otherwise, the service time of the leading mail carrier is s_j. We have

$$s_{ld}(\beta) = \begin{cases} s_j & \text{if } T(\beta) < s_{tr} + \alpha\beta(d_j - loc_{ld}) + \alpha(loc_T - loc_{tr}) + (1-\beta)(d_j - loc_{ld}), \\ 0, & \text{otherwise.} \end{cases} \tag{D.6}$$

On the other hand, if $loc_{ld} > loc_T$, i.e., the truck is not with the leading mail carrier, β can only be 0. In this situation, we have the same four cases for the location of the trailing mail

carrier. This yields

$$loc_{tr}(0) = \begin{cases} loc_{tr}, & \text{if } s_{tr} \geq T(0), \\ loc_{tr} + (T(0) - s_{tr})/\alpha, & \text{if } s_{tr} < T(0) \leq s_{tr} \\ & +\alpha(loc_T - loc_{tr}), \\ loc_T + T(0) - (s_{tr} + \alpha(loc_T - loc_{tr})), & \text{if } s_{tr} + \alpha(loc_T - loc_{tr}) < T(0) \leq \\ & s_{tr} + \alpha(loc_T - loc_{tr}) + (d_j - loc_T), \\ d_j, & \text{otherwise.} \end{cases} \tag{D.7}$$

For the location of the truck, depending on if the trailing mail carrier catches the truck, $loc_T(0) = \max\{loc_T, loc_{tr}(0)\}$. For the service time of the leading and trailing carriers, they are the same as the case of $loc_T = loc_{ld}$. We have built a recursion relationship for all states and hence the DP formulation is complete.

Appendix E: MIP Formulation for the General Case of the PMCP

We present an MIP formulation for the general case. The MIP formulation discretizes the travel time between adjacent stops. Thus, it gives an approximate solution based on the level of discretization. However, in practice, such a technique often provides satisfactory performance by carefully selecting the level of discretization.

In this MIP formulation, stops 0 and $N+1$ are the start and end points of the truck route, respectively. Both have a service time of 0. $J_{stop} = \{0, 1, 2, \ldots, N+1\}$ is the set of service stops. For $0 \leq j \leq N$, we evenly divide the interval between stops j and $j+1$ into P sub-intervals. Thus, there are $P-1$ new break points between stops j and $j+1$. These break points share a service time of 0. We relabel these stops and break points in a set $J_{total} = \{0, 1, 2, \ldots, P(N+1)\}$. J^{-}_{total} is J_{total} minus the last point $P(N+1)$. Note that stop j is labeled as point jP in J_{total}. The required service time s_j of stop j is associated with point jP as well. $I = \{A_1, A_2, \ldots, A_k\}$ is the set of mail carriers. L_j is the number of minutes it takes to drive the truck from point 0 (start point) to point j, and M is the "big M" commonly used as a large constant in MIP contexts.

We introduce the following non-negative variables: ct is the completion time. $A_{i,j}$ is the arrival time of mail carrier i at point j. $D_{i,j}$ is the departure time of mail carrier i at point j. c_j is the departure time at point j for the truck. Also, we need the following binary variables: $x_{i,jP} = 1$ if mail carrier i services stop j (point jP); otherwise $x_{i,jP} = 0$. When a mail carrier

i leaves point j in the truck, $y_{i,j} = 1$. Otherwise, $y_{i,j} = 0$. Now, we are ready to write the MIP formulation which is referred to as MIPE.1.

$$\text{(MIPE.1) Min} \quad ct \tag{E.1}$$

$$\text{Subject to} \quad \sum_{i \in I} x_{i,jP} = 1 \qquad \forall j \in J_{stop} \tag{E.2}$$

$$A_{i,j+1} \geq D_{i,j} + (L_{j+1} - L_j)[y_{i,j} + \alpha(1 - y_{i,j})] \qquad \forall i \in I, j \in J^-_{total} \tag{E.3}$$

$$D_{i,jP} \geq A_{i,jP} + s_j x_{i,jP} \qquad \forall i \in I, j \in J_{stop} \tag{E.4}$$

$$c_j - D_{i,j} \geq M(1 - y_{i,j}) \qquad \forall i \in I, j \in J_{total} \tag{E.5}$$

$$c_j - D_{i,j} \leq M(1 - y_{i,j}) \qquad \forall i \in I, j \in J_{total} \tag{E.6}$$

$$ct \geq A_{i,P(N+1)} \qquad \forall i \in I \tag{E.7}$$

$$A_{i,0} = 0 \qquad \forall i \in I \tag{E.8}$$

$$A_{i,j}, D_{i,j} \geq 0 \qquad \forall i \in I, j \in J_{total} \tag{E.9}$$

$$c_j \geq 0 \qquad \forall j \in J_{total} \tag{E.10}$$

$$x_{i,jP} \in \{0, 1\} \qquad \forall i \in I, j \in J_{stop} \tag{E.11}$$

$$y_{i,j} \in \{0, 1\} \qquad \forall i \in I, j \in J_{total} \tag{E.12}$$

In MIPE.1, the objective function (E.1) involves minimizing the completion time. Constraints (E.2) require exactly one mail carrier to serve stop j (point jP). Constraints (E.3) force the arrival time of mail carrier i at point $j + 1$ to be the sum of the departure time from point j of mail carrier i, the amount of time spent walking between points j and $j + 1$ of mail carrier i, and the amount of time spent in the truck between points j and $j+1$ of mail carrier i. Constraints (E.4) calculate the departure time of mail carrier i at a stop j by enforcing the service time if mail car-

rier i is assigned to stop j. Constraints (E.5) and (E.6) together ensure that if a mail carrier leaves point j in the truck, the departure time of the carrier at point j is the same as that of the truck. Constraints (E.7) make sure that the completion time is greater than or equal to the completion times of the mail carriers. Constraints (E.8) set the starting time as 0. Constraints (E.9), (E.10), (E.11) and (E.12) are non-negative and binary constraints.

In MIPE.1, $P-1$ points are introduced to discretize the travel time between two consecutive service stops. Including the original service stops, there are $(N+1)*P+1$ points. By imposing constraints (E.3), (E.5), and (E.6), we require a mail carrier to get in the truck only at one of these points. Once the mail carrier is in the truck, the carrier must stay in it until the next point. We want to point out that MIPE.1 is an approximate model. However, when P increases, the approximation improves. When P is an infinite number, MIPE.1 becomes an infinite IP model. Although it gives exact solutions, the resulting MIPE.1 is impractical and computationally challenging. Nonetheless, a reasonable number P can already provide a good approximation in practice.

Bibliography

[1] A. Alvarez and P. Munari. An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. *Computers & Operations Research*, 83:1–12, 2017.

[2] Amazon. How your package gets from amazon's warehouse to your front door, Jan 2019. URL https://www.aboutamazon.co.uk/news/operations/how-your-package-gets-from-amazons-warehouse-to-your-front-door. accessed on 2022-06-27.

[3] M. A. Andresen and N. Malleson. Police foot patrol and crime displacement. *Journal of Contemporary Criminal Justice*, 30(2):186–199, 2014.

[4] U. Basaninyenzi. Social dimensions of climate change. URL https://www.worldbank.org/en/topic/social-dimensions-of-climate-change. accessed on 2022-06-27.

[5] H. Ben Amor, J. Desrosiers, and J. M. Valério de Carvalho. Dual-optimal inequalities for stabilized column generation. *Operations Research*, 54:454–463, 2006.

[6] Z. Blank. Ups orion new update - how can this technology boost driver efficiency?, Jul 2021. URL https://www.getstraightaway.com/blog-posts/ups-orion-new-update-how-can-this-technology-boost-driver-efficiency. accessed on 2022-06-27.

[7] A. A. Braga. Policing crime hot spots. *Preventing Crime*, page 179–192, 2007.

[8] A. A. Braga, B. Turchan, A. V. Papachristos, and D. M. Hureau. Hot spots policing of small geographic areas effects on crime. *Campbell Systematic Reviews*, 15, 2019.

[9] A. Buck. 57 amazon statistics to know in 2022, Apr 2022. URL https://landingcube.com/amazon-statistics/. accessed on 2022-06-27.

[10] K. Buhrkal, A. Larsen, and S. Ropke. The waste collection vehicle routing problem with time windows in a city logistics context. *Procedia - Social and Behavioral Sciences*, 39: 241–254, 2012.

[11] D. o. J. Bureau of Justice Assistance. Understanding community policing: A framework for action, 1994. URL https://www.ojp.gov/pdffiles/commp.pdf. accessed on 2022-06-28.

[12] I. Çapar, B. B. Keskin, and P. A. Rubin. An improved formulation for the maximum coverage patrol routing problem. *Computers & Operations Research*, 59:1–10, 2015.

[13] I.-M. Chao, B. L. Golden, and E. A. Wasil. The team orienteering problem. *European Journal of Operational Research*, 88(3):464–474, 1996.

[14] S. S. Chawathe. Organizing hot-spot police patrol routes. *2007 IEEE Intelligence and Security Informatics*, 2007.

[15] H. Chen, T. Cheng, and S. Wise. 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, 2015.

[16] H. Chen, T. Cheng, and S. Wise. Developing an online cooperative police patrol routing strategy. *Computers, Environment and Urban Systems*, 62:19 – 29, 2017.

[17] P. A. Chircop, T. J. Surendonk, M. H. van den Briel, and T. Walsh. A branch-and-price framework for the maximum covering and patrol routing problem. *Lecture Notes in Management and Industrial Engineering*, page 59–80, 2021.

[18] M.-A. Coindreau, O. Gallay, and N. Zufferey. Vehicle routing with transportable resources: Using carpooling and walking for on-site services. *European Journal of Operational Research*, 279:996–1010, 2019.

[19] T. G. Crainic, N. Ricciardi, and G. Storchi. Advanced freight transportation systems for congested urban areas. *Transportation Research Part C: Emerging Technologies*, 12:119–137, 2004.

[20] K. M. Curtin, K. Hayslett-McCall, and F. Qiu. Determining optimal police patrol areas with maximal covering and backup covering location models. *Networks and Spatial Economics*, 10(1):125–145, 2007.

[21] G. Dalla Chiara and A. Goodchild. Do commercial vehicles cruise for parking? empirical evidence from seattle. *Transport Policy*, 97:26–36, 2020.

[22] R. Dewil, P. Vansteenwegen, D. Cattrysse, and D. Van Oudheusden. A minimum cost network flow model for the maximum covering and patrol routing problem. *European Journal of Operational Research*, 247:27–36, 2015.

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

[24] DHL. Dhl and reef technology launch pilot to use ecofriendly cargo bikes for deliveries in downtown miami, May 2020. URL `https://www.dhl.com/us-en/home/press/press-archive/2020/dhl-and-reef-technology-launch-pilot-to-use-ecofriendly-cargo-bikes-for-deliveries-in-downtown-miami.html`. accessed on 2022-06-27.

[25] C. Fikar and P. Hirsch. A matheuristic for routing real-world home service transport systems facilitating walking. *Journal of Cleaner Production*, 105:300–310, 2015.

[26] M. Folger. Electrify ups: A zero emission shipping strategy, Oct 2021. URL https://environmentamerica.org/blogs/environment-america-blog/ame/electrify-ups-zero-emission-shipping-strategy. accessed on 2022-06-27.

[27] B. Gilson. Business transformation 2010 and beyond: a national agreement between royal mail & the communication workers' union, Mar 2010. URL https://www.cwusouthmidspostal.org/wp-content/uploads/2018/02/Business-Transformation-2010-Beyond-Agreement.pdf. accessed on 2022-06-27.

[28] B. L. Golden, S. Raghavan, E. A. Wasil, et al. *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43. Springer, 2008.

[29] K. Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.

[30] J. Holguin-Veras et al. Impacts of freight parking policies in urban areas: the case of new york city. 2016.

[31] 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*, 48:215–232, 2012.

[32] J. Kinable, W.-J. van Hoeve, and S. F. Smith. Optimization models for a real-world snow plow routing problem. *Integration of AI and OR Techniques in Constraint Programming*, page 229–245, 2016.

[33] R. Kistner. The robust vehicle routing problem: An application of the traveling salesman problem with center. Master's thesis, University of Maryland College Park, 2010.

[34] C. S. Koper. Just enough police presence: Reducing crime and disorderly behavior by optimizing patrol time in crime hot spots. *Justice Quarterly*, 12(4):649–672, 1995.

[35] R. E. Korf. A complete anytime algorithm for number partitioning. *Artificial Intelligence*, 106:181–203, 1998.

[36] C. Kuntze, A. Martin, C. Regnier, and I. Silva. Deliver on time or pay the fine: Speed and precision as the new supply-chain drivers, Dec 2020. URL https://www.mckinsey.com/business-functions/operations/our-insights/deliver-on-time-or-pay-the-fine-speed-and-precision-as-the-new-supply-chain-drivers. accessed on 2022-06-27.

[37] Lake Oswego Review. Ask a cop: How much mileage is put on a police car?, Dec 2014. URL https://pamplinmedia.com/lor/48-news/244840-112074-ask-a-cop-how-much-mileage-is-put-on-a-police-car. accessed on 2022-06-28.

[38] R. C. Larson. *Urban Police Patrol Analysis*, volume 28. MIT Press, 1972.

[39] J. Leigh, S. Dunnett, and L. Jackson. Predictive police patrolling to target hotspots and cover response demand. *Annals of Operations Research*, 283(1):395–410, 2019.

[40] C. Lin. A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources. *Computers & Operations Research*, 38:1596–1609, 2011.

[41] A. Lipowski and D. Lipowska. Traveling salesman problem with a center. *Physical Review E*, 71(6):067701, 2005.

[42] L. Lozano, D. Duque, and A. L. Medaglia. An exact algorithm for the elementary shortest path problem with resource constraints. *Transportation Science*, 50:348–357, 2016.

[43] P. Matl, R. F. Hartl, and T. Vidal. Workload equity in vehicle routing problems: A survey and analysis. *Transportation Science*, 52(2):239–260, 2018.

[44] F. McLeod, T. Cherrett, T. Bektas, J. Allen, A. Martinez-Sykora, C. Lamas-Fernandez, O. Bates, K. Cheliotis, A. Friday, M. Piecyk, and S. Wise. Quantifying environmental and financial benefits of using porters and cycle couriers for last-mile parcel delivery. *Transportation Research Part D: Transport and Environment*, 82:102311, 2020.

[45] J. Milan. Delivering difference: Mailing services uses e-bikes to move mail at uw, Nov 2018. URL `https://green.uw.edu/blog/2018-11/delivering-difference-mailing-services-uses-e-bikes-move-mail-uw`. accessed on 2022-06-27.

[46] A. Mor and M. G. Speranza. Vehicle routing problems over time: A survey. *Annals of Operations Research*, pages 1–21, 2022.

[47] B. Morgan. 101 companies committed to reducing their carbon footprint, Dec 2021. URL `https://www.forbes.com/sites/blakemorgan/2019/08/26/101-companies-committed-to-reducing-their-carbon-footprint/?sh=6002869f260b`. accessed on 2022-06-27.

[48] C. C. Price. *Applications of operations research models to problems in health care*. PhD thesis, University of Maryland College Park, 2009.

[49] V. Pureza, R. Morabito, and M. Reimann. Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the vrptw. *European Journal of Operational Research*, 218:636–647, 2012.

[50] S. Reed, A. M. Campbell, and B. W. Thomas. Does parking matter? the impact of search time for parking on last-mile delivery optimization. *arXiv preprint arXiv:2107.06788*, 2021.

[51] S. Reed, A. M. Campbell, and B. W. Thomas. Impact of autonomous vehicle assisted last-mile delivery in urban to rural settings. *Transportation Science*, 2022.

[52] S. Reed, A. M. Campbell, and B. W. Thomas. The value of autonomous vehicles for last-mile deliveries in urban environments. *Management Science*, 68:280–299, 2022.

[53] L. Rosencrance. The importance of last-mile delivery to logistics, Jun 2021. URL https://www.techtarget.com/searcherp/feature/The-importance-of-last-mile-delivery-to-logistics. accessed on 2022-06-27.

[54] S. R. Sacks. Optimal spatial deployment of police patrol cars. *Social Science Computer Review*, 18(1):40–55, 2000.

[55] D. Sanchez. Cspy: A python package with a collection of algorithms for the (resource) constrained shortest path problem. *Journal of Open Source Software*, 5:1655, 2020.

[56] G. Senarclens de Grancy and M. Reimann. Vehicle routing problems with time windows and multiple service workers: A systematic comparison between aco and grasp. *Central European Journal of Operations Research*, 24:29–48, 2014.

[57] D. Sweeney. Why do cops get to take their patrol cars home, and how is it regulated?: You asked, we answer, Nov 2019. URL https://www.sun-sentinel.com/news/sound-off-south-florida/fl-ne-sosf-police-take-home-patrol-cars-20191108-b2s7thkvu5gtpl5ta2jy4jxy5m-story.html. accessed on 2022-06-28.

[58] M. Takamiya and T. Watanabe. Planning high responsive police patrol routes with frequency constraints. *Proceedings of the 5th International Confernece on Ubiquitous Information Management and Communication - ICUIMC '11*, 2011.

[59] C. Tilk, A.-K. Rothenbächer, T. Gschwind, and S. Irnich. Asymmetry matters: Dynamic half-way points in bidirectional labeling for solving shortest path problems with resource constraints faster. *European Journal of Operational Research*, 261:530–539, 2017.

[60] P. Toth and D. Vigo. *The Vehicle Routing Problem*. SIAM, Philadelphia, USA, 2002.

[61] P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications*. SIAM, Philadelphia, USA, 2014.

[62] UPS. Smart, sustainable solutions help ups deliver: About ups, Jun 2018. URL https://about.ups.com/mx/en/social-impact/environment/climate/smart--sustainable-solutions-help-ups-deliver.html. accessed on 2022-06-27.

[63] USPS. Fleet of feet - u.s. postal facts, Mar 2022. URL https://facts.usps.com/fleet-of-feet/. accessed on 2022-06-27.

[64] N. Williams and D. Murray. An analysis of the operational costs of trucking: 2020 update, 2020. URL https://truckingresearch.org/wp-content/uploads/2020/11/ATRI-Operational-Costs-of-Trucking-2020.pdf. accessed on 2022-06-27.

[65] K. E. Wuschke, M. A. Andresen, P. J. Brantingham, C. Rattenbury, and A. Richards. What do police do and where do they do it? *International Journal of Police Science & Management*, 20(1):19–27, 2017.