# VEHICLE ROUTING MODELS IN PUBLIC SAFETY AND HEALTH CARE


by

RONG LI

BURCU B. KESKIN, COMMITTEE CHAIR

JOHN MITTENTHAL

CHARLES SCHMIDT

SHARIF MELOUK

YINGYAN LOU


A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Information Systems,
Statistics and Management Science
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2012

ABSTRACT

Routing related costs constitute a significant portion of the overall logistics costs in most service industries. Private companies are continuously striving to reduce their vehicle routing costs to maintain a better standing in the competitive business world. In contrast, the public sector has not paid enough attention to its vehicle routing efficiency. Vehicle routing inefficiencies have resulted in wasting resources. Due to the recent funding cuts and economic hardship, public agencies need to improve their vehicle routing efficiencies.

To aid public agencies in resolving inefficiencies in their operations, we propose some challenging vehicle routing problems in the public sector through mathematical modeling. To achieve this goal, in this dissertation, we study vehicle routing problems in two fields: i) vehicle routing model in public safety, i.e., how state troopers can patrol more efficiently and effectively on the roadways; and ii) vehicle routing model in health care, i.e., how caregivers are assigned to patients with home care needs and how they schedule their visit sequences.

In the context of public safety, we present two models: i) a single-period, single-depot team orienteering problem with time windows and ii) a multi-period, multi-depot team orienteering problem with time windows. In the context of home health care, we present a multi-period, multi-depot vehicle routing problem with constraints specific to the health care industry. All of these models are mixed integer, and considered as computationally intractable. We solve them using either heuristics (local search, tabu search, simulated annealing) or decomposition method (column generation).

Model one in public safety finds efficient patrolling plan with one single state trooper post. Model two in public safety improves the coverage of the roadway by allowing multiple state trooper posts. And, model three in home health care demonstrates improvements over the current practice with respect to the traveling cost and workload balance, and answers the question whether to invest in purchasing centrifuges.

# DEDICATION

This dissertation is dedicated to my beloved father Wanqin Li, mother Jinfang Xiong, and fiancé Yan Li.

## LIST OF ABBREVIATIONS AND SYMBOLS

| | | |
|---|---|---|
| $i$ | = | Customer vertex index |
| $\mathcal{N}$ | = | Customer vertex set |
| $\mathcal{V}$ | = | Customer and depot vertex set |
| $k$ | = | Car index |
| $\mathcal{K}$ | = | Car set |
| $e_i, l_i$ | = | Earliest and latest time window of customer $i$ |
| $\mathcal{E} = \{(i,j):\ i,\ j \in \mathcal{V}, i \neq j\}$ | = | Edge set |
| $t_{ij}$ | = | Travel time between $i,\ j \in \mathcal{V}, i \neq j$ |
| $w_i$ | = | The severity level of customer $i$ |
| $t$ | = | Period index |
| $\mathcal{T}$ | = | Period set |
| $j$ | = | Request index |
| $\mathcal{J}$ | = | Service set |
| $q_j$ | = | Caregiver type of request $j$ |
| $s_j$ | = | Service time of request $j$ |
| $o$ | = | The numbering index of a visit of a request |
| $f_{jo}$ | = | The frequency of request $j$ for its $o$th visit |

# ACKNOWLEDGMENT

First and foremost, I want to thank my father Wanqin Li and my mother Jinfang Xiong. It is they who brought me to this wonderful world; it is they who shaped my personality; it is they who taught me the fundamental qualities to be successful in the future, including perseverance, diligence, courage, patience, and curiosity; and it is they who have always been there to protect me and support me unconditionally, no matter I am up or down. They encourage me to conquer any obstacles in my study and endure any pain from living in a foreign country alone.

I am very thankful to my advisor, Dr. Burcu Keskin. Someone said that a good advisor is like a parent. I am fortunate to have Dr. Keskin as my advisor who treats me like her own daughter. Every time when I could not make any progress, she always knows if I am slacking off or stuck. She "whips" me when I am slacking off, and she helps me when I am stuck. This is a very good combination of carrot and stick. I am also very thankful to all my other committee members. They not only helped me with my research, but also helped me with other important things in my life. Dr. Mittenthal and Dr. Melouk helped me with job hunting, teaching skills, and resolve my frustrations; Dr. Schmidt helped me solve some technical questions; Dr. Lou also helped me with job hunting and teaching.

I would also like to thank all of my local friendship family members in Tuscaloosa. They invited me to their homes to experience American lives, which I fully felt the so-called "Southern Hospitality". In the past four years, they have helped me tremendously immerse into the society, understand the culture, and improve my communication skills. With them, even though I am alone in US, I am never lonely.

Last but not least, I want to thank my fiancé and soul mate Yan Li, who shares all my happiness and softens my sorrows. Thanks to him, I am full of hope for life, schooling, and future job.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

# INTRODUCTION

Routing efficiency can significantly affect the profitability of both public and private sector organizations. In the public sector, services including the postal service, garbage collection, and health care would benefit from better routing practices. Whereas, in the private sector, industries including airlines, locomotives, parcel deliveries, and telecom networks heavily focus on routing efficiency. Private companies are profit-driven, thus more sensitive towards good routing plans than public agencies. Brutal market competition further adds momentum to this pursuit by private companies. On the contrary, since public agencies do not focus on profit maximization, better utilization of resources was not a primary focus. Specifically, their routing practices were negatively influenced by the lack of such a focus. Due to poor economic condition, the government has issued a series of spending cuts in the public agencies at both federal and state levels. The spending cuts in public agencies indicate that public agencies and not-for-profit institutions now need to shift their focus towards operational excellency and efficiency, including routing operations.

Vehicle routing is significant because the costs of vehicle routing are substantial. According to the 22nd Annual State of Logistics Report released in June 2011, the U.S. business logistics system cost hit $1.2 trillion in 2010, accounting for 8.3% of the national gross domestic product (Burnson, 2011). The logistics costs consist of transportation, administration, and inventory costs, in which the transportation costs account for more than 50% (Burnson, 2011). From 2009 to 2010, the transportation costs increased by more than 10% due to higher freight volumes, fuel surcharges, and for some modes, rate hikes (Burnson, 2011). According to Toth and Vigo (2002), with the utilization of computerized methods for transportation, the resulted savings are as significant as 5% to 20%.

Due to the aforementioned reason, the vehicle routing problems attracted attention both from academia and industry. Practitioners and academicians are constantly seeking new transportation modes, sequencing rules, and efficient heuristics to find better vehicle routing solutions. Vehicle routing problem is defined as the problem of designing least-cost delivery routes from a depot to a set of geographically scattered customers, subject to side constraints (Laporte, 2009). The first vehicle routing problem was done by Dantzig and Ramser (1959), who solved a truck dispatching system. Another classic work was published in 1964, in which Clarke and Wright (1964) developed a heuristic algorithm to improve the Dantzig and Ramser (1959) approach. Depending on the diversity of the constraints encountered in real-life applications, vehicle routing problems can be classified as capacitated vehicle routing problem (CVRP), distance-constrained vehicle routing problem (DCVRP), vehicle routing problem with backhauls (VRPB), vehicle routing problem with time windows (VRPTW), and vehicle routing problem with pickup and delivery (VRPPD) (Desrochers et al., 1990; Toth and Vigo, 2002). All of these listed categories are classical vehicle routing problems, in which each route starts and ends at the depot, each customer is visited exactly once by a single vehicle, and the objective is to minimize the total routing cost. To get an overview of vehicle routing problems, we refer to Laporte and Osman (1995) for a bibliography, Desrochers et al. (1990) for a classification of the problem, and Laporte (1992) for the algorithms used in the previous studies. In addition, three books (Golden et al., 2008; Laporte, 1988; Toth and Vigo, 2002) provide more systematic and comprehensive discussions on vehicle routing problems.

Even though classical vehicle routing problems have been studied extensively, Golden et al. (2008) pointed out there are new challenges awaiting to be solved, including real-time vehicle routing, multi-period, multi-objective, the orienteering problem, new health care topics (i.e., blood logistics, ambulance service), and disaster relief. Most of these topics are from the public sector, such as health care, disaster relief, and police patrolling. As a consequence, there has been a surge of interest in vehicle routing in the public sector recently. The public sector differs from the private sector in its goal, which is not profit-driven but instead targets at providing people with good service while minimizing the overall cost. With multiple problems, the e-constraint approach is a common technique that basically keeps the most important objective and converts the others to relevant constraints. Hence, the classical vehicle routing problem's objective of cost minimization or benefit maximization, along with some constraints (i.e., budget limit, service level, equity, etc.), is appropriate for public agencies.

Realizing the importance of vehicle routing and following new trends, we focus on two important areas: i) public safety and law enforcement and ii) health care. Specifically, this dissertation addresses the following vehicle routing problems:

**Public safety and law enforcement** It is believed that concentrated traffic enforcement (e.g. state trooper patrolling) can be helpful in reducing the number of crashes and discouraging dangerous behavior (Steil and Parrish, 2009). However, since there are not enough resources (state troopers, operational patrol routing funding, etc.) to cover every possible accident site, we research how to effectively deploy state troopers to reduce traffic accidents.

**Health care** Within health care domain, we specifically focus on the operations of home health care agencies. Home care agencies satisfy the needs of their patients by assigning appropriate caregivers during certain prescribed times over a recovery horizon while keeping the routing and staffing costs in check. Scheduling home caregivers is not trivial due to interwoven health care related constraints. Apart from the classical vehicle routing constraints, the additional complexity stems from the unique service characteristics of home health care: qualification-need match, blood test, caregiver-patient continuity, workload balance, synchronized visit requests by one patient, and sequential visit requests by one patient in one day. The research questions are i) how patients should be assigned to caregivers and ii) how caregivers should route so that the total traveling cost is minimized and the service level is guaranteed.

These two topics have several similarities. First, we solve the vehicle routing problems of both topics, following the same line of theory and literature. Second, both topics have the objective of benefit maximization. The objective of the first topic is to maximize the effective coverage time. Even though the objective of the second topic is to minimize the total traveling cost, with a fixed revenue, it is equivalent to maximizing the benefit. Third, both topics have time window considerations. The time windows of the first topic are when hot spots are critical, and time windows of the second topic are when patients need services. Fourth, both topics have resource limitations. The transportation department has a fixed number of available state troopers, and the agency has a fixed number of available caregivers.

## DISSERTATION OBJECTIVES AND CONTRIBUTIONS

The current routing practices in these problems are ad-hoc based, thus inevitably causing a waste of resources and even poor quality of routing execution, hence not covering a stretch of road that could have covered or not serving a patient that could have been visited. More specifically, in patrolling and law enforcement, state troopers decide where and when to visit based on their personal experiences or even preferences; whereas, in home health care, there is a central dispatcher assigning patients to caregivers at the master-level according to her own judgement, and caregivers decide their own visiting sequences to serve their assigned patients at the operational level. In this research, we aim to improve vehicle routing efficiency by utilizing operations research techniques to model these complicated problems and strive for the optimal schedules and routes. The specific objectives and the expected contributions are as follows:

- to improve the decision-making process by streamlining vehicle routing operations for state troopers and the central dispatcher of the home health care agency,
- to develop optimization models to schedule and route state troopers and home caregivers,
- to design the state-of-the-art algorithms to solve these challenging vehicle routing models efficiently and effectively, thus contributing to the current vehicle routing literature,

- to introduce service-related performance measures other than the cost/benefit objective itself, and
- to compare our approach with the current practice and to offer recommendations and managerial insights to improve the vehicle routing efficiencies for both the transportation departments and the home care agencies.

## DISSERTATION FORMAT

This dissertation follows a three-article format.

- The first article solves a single-period, single-depot team orienteering problem with time windows in the context of public safety and law enforcement.
- The second article extends the first one by solving a multi-period, multi-depot team orienteering problem with time windows.
- The third article solves a multi-period, multi-depot version of the vehicle routing problem in the context of health care, and it is computationally more sophisticated due to the more convoluted constraints specific to home health care.

These three articles are inter-related in that the first article lays a foundation for the second one, and the third article theoretically advances the first two articles.

The first article, named "Analysis of an integrated maximum covering and patrol routing problem", introduces a new application area of the team orienteering problem with time windows, which is a variant of the vehicle routing problem by allowing customers not visited. The paper targets at improving the safety level of the road network in Alabama. CRDL (CARE Research and Development Laboratory) in the University of Alabama provides us with the data about the accident-prone locations, namely hot spots. Each county has a state trooper post, where its state troopers start patrolling at the beginning of a shift and come back in the end, and each day is divided to three shifts of equal length. With the presence of state troopers on the road, drivers tend to slow down and drive more safely, resulting in less accidents. However, there are not enough state troopers to cover all of the hot spots, so the goal is to dispatch the available state troopers to cover the hot spots as effectively as possible. With this problem setup, our patrol routing problem can be solved independently county by county and shift by shift. We present a mixed integer linear programming formulation, prove its NP-hardness, and solve it using heuristics, including both local search and tabu search. We run our solution algorithms on both randomly generated data and real life data. Based on our experimentations, we give suggestions to the decision makers on how to improve state trooper patrolling efficiency. To help them evaluate our suggested vehicle routing plans, we not only report the objective, but also report two performance measures that they value: the percentage of hot spot covered and the percentage of coverage length. Jefferson, Mobile, and Tuscaloosa areas are our targeted areas due to their high number of accidents. In these three areas, we observe that our suggested plan has hot spot coverage percentages quite close to 100% and the objective coverage percentage above 85% in most instances.

The second article, named "Bi-criteria dynamic location-routing problem for patrol coverage", extends the previous article. The motivation comes from a counter-intuitive finding in the previous work. We observe that the objective coverage reaches a plateau in some instances; no matter how we increase the resources. Even if we have unlimited resources, the coverage of hot spots could not reach 100%. This phenomenon is attributed to the assumption of the single state trooper post. There are some hot spots which are located too far away from a state trooper post such that the time windows of the hot spots have already elapsed before a state trooper is able to even reach them by directly traveling from the state trooper post. To improve the coverage time, we relax this assumption to allow for the existence of multiple state trooper posts. The resulting problem is in essence a combination of the team orienteering problem with time windows and the facility location problem. We present an updated mixed integer linear programming formulation, decompose the problem to two subproblems, which are the multi-depot version of the previous problem and the facility location problem, and solve them iteratively. To improve the solution quality, we apply simulated annealing to our decomposition solution approach, which strengths the ability of finding the global optimum by accepting worsen solutions with some probabilities. Since this article is an extension of the preceding one, we compare our new solution with that of the previous work. We compare them based on both uniformly distributed and clustered distributed HSs. The extended model is moderately better than the base model for uniformly distributed HSs, and the extended model is significantly better than base model for clustered distributed HSs. In practice, the extension leads to much better coverage. Better coverage indicates that roadways are safer.

The third article, named "A multi-period home care scheduling problem with work balance", switches gears from public safety to health care. We consider the problem of assigning caregivers to patients and scheduling caregivers' visit sequences to patients' homes subject to a host of constraints. The constraints include both the classical vehicle routing problem constraints and several additional constraints unique to home care. The first group of constraints forms feasible routes by sending a group of caregivers of different qualifications to visit patients with different needs, which start and end at caregivers' homes or the home care agency subject to the work hour limit. The second group adds the following five complicating situations.

- Blood test: for the diagnosis and treatment of certain diseases, blood test is usually necessary, but blood samples only remain viable for 2 hours and sometimes they can only be analyzed in the specialized labs which may not be close to the patients' homes, thus significantly affecting the design of routes and schedules of the caregivers.

- Synchronization: certain patients' needs require services from multiple caregivers simultaneously.

- Precedence: certain patients' needs require sequential services from caregivers of different specializations.

- Workload balance: the home care agency wants to increase its employee satisfaction by balancing the caregivers' workloads. And
- continuity: patients usually prefer being treated by the same caregivers, and caregivers also prefer treating the same patients.

For the home care scheduling problem with work balance, we are the first to present a mixed integer linear programming formulation. We apply a column generation based decomposition method, which is widely used in the vehicle routing problem and the crew scheduling problem literature, and successfully solve our real problem with hundreds of patients and dozens of caregivers faced by the home care agency under our study. Compared with the home care agency's current practice, our solution reduces the routing cost significantly in the first several weeks. Moreover, our solution reduces the the heaviest workload of each caregiver from more than 40 patients to 25 patients per week. In addition, we compare the solutions when centrifuges are available with the ones when they are not available. The results reveal that the savings are not sufficient to cover the investment cost, so the home care agency should not purchase centrifuges.

This dissertation concludes with a summary of results and managerial insights developed through this research. The conclusion section also provides potential extensions to the work done in this dissertation.

# 1: ANALYSIS OF AN INTEGRATED MAXIMUM COVERING AND PATROL ROUTING PROBLEM

## 1.1 INTRODUCTION

Traffic accidents pose a great danger to passengers' lives. In 2009, 33,963 people died in traffic crashes in the United States, an 8.9% decline from 2008 and the lowest total since 1954 (National Highway Traffic Safety Administration (NHTSA), 2010). Even though fatality rates continue to drop in the United States, the number of fatalities is still significant. Furthermore, the economic impact of motor vehicle crashes on U.S. roadways is still noteworthy. The NHTSA estimates this cost as $230.6 billion per year (nearly 2.3 percent of the nation's gross domestic product) or an average of $820 per person in the country (Blincoe et al., 2002). Thus, it is of humanitarian and economic importance to reduce traffic accidents.

It is believed that concentrated traffic enforcement has a positive impact in reducing the number of crashes and discouraging dangerous behavior (Steil and Parrish, 2009). One such example, the NHTSA-sponsored "Click it or Ticket" program, uses a combination of publicity and increased law enforcement to educate and motivate the public. Another program, "Targeting Aggressive Cars and Trucks," sponsored by the Federal Motor Carriers Safety Administration (FMSCA, 2008), encourages the participating states to identify additional law enforcement and publicity strategies that will deter aggressive driving. Due to limited resources, a primary concern of public safety officials is the *effective use of patrol cars and state troopers* in reducing traffic accidents. A typical method for state troopers is to patrol "hot spots," that is, certain locations of highways with high frequencies of crashes over a certain time period. These locations are often associated with a particular type of crash (e.g., excessive crashes caused by speed or DUI). Furthermore, hot spots vary with respect to the day of week and time of day, i.e., a particular highway location may be a hot spot on a particular day and time but not "hot" at other times.

With this motivation, given identified hot spots on mile-posted highways, we focus on building effective state trooper patrol routes with maximum hot spot coverage. This problem has similarities to **the orienteering problem (OP)** (Tsiligirides, 1984; Feillet et al., 2005), also known as the Selective Traveling Salesman Problem (STSP), that consists of finding a circuit that maximizes collected profit such that travel costs do not exceed a preset value $C$. For our problem, the service time at a hot spot can be viewed as the "*profit*" whereas the shift duration is equivalent to setting a value for $C$. However, due to time windows of hot spots, we have an "*expiration time*" on the "*profits*." Furthermore, we consider routing multiple cars simultaneously. Therefore, our problem is related to the Team Orienteering Problem with Time Windows (TOPTW), a variant of OP. In the TOPTW, the goal is to maximize the total profit by a fixed number of routes such that the locations are visited within a time window and the maximum tour length is limited. The main difference between our problem and the TOPTW is that we do not have a fixed "*profit*" associated with each location. The collected profit depends on the service time which could be as low as one minute or as large as the length of the time window (up to 270 minutes).

This property necessitates a novel solution approach to the problem. For this purpose, we develop a mixed integer programming formulation. For real data, unfortunately, the problem is not solvable computationally using state-of-the-art commercial solver, CPLEX 12.1[1]. In fact, in the appendix, we prove that our problem belongs to the class of NP-hard problems as the OP (Golden et al., 1987). Therefore, we focus on local search– and tabu search– based heuristic approaches that provide good quality solutions in short periods of time. Since this problem needs to be solved over a number of state trooper post regions, several days, and many shifts, having fast and effective heuristic approaches actually is a requirement for the applicability of the solutions by practitioners. As it is not possible to cover all of the hot spots

[1]CPLEX is a trademark of IBM.

with given resources, we also provide additional service measures including the percentage of number of hot spots covered and percentage of coverage length based on the outcome of the heuristics. These service measures provide additional insights into the solutions and help in evaluating the constraints related to the number of state trooper cars and patrol duration.

To summarize, this paper is unique in the sense that it considers the integrated optimization of strategic crash covering and patrol routing problems while designing an efficient operating plan for state troopers. Its formulation is a methodological contribution to the current literature. Furthermore, the problem-specific heuristic approaches– local and tabu searches– help decision-makers act fast and rationally to ensure traffic law enforcement.

The remainder of this paper is structured as follows. In Section 1.2, we present the literature review. In Section 1.3, we present the general mathematical model, including necessary assumptions and notation. In Section 1.4, we present the analysis of the problem and the solution approaches based on the characteristics of the problem. In Section 1.5, we discuss the computational results based on randomly generated data and real data. Finally, in Section 1.6, we provide our conclusions, recommendations, and future work.

## 1.2 LITERATURE REVIEW

Our research builds on the assumption that it is possible to identify hot spots, where accidents are more likely to happen. Most of the literature on accident analysis and prevention focuses on methods identifying hot spots (Anderson, 2006; Cheng and Washington, 2005; Chen and Quddus, 2003; Gatrell et al., 1996; Miranda-Moreno et al., 2007; McCullagh, 2006; Steil and Parrish, 2009). However, our focus is not on hot spot identification. To identify hot spots, we utilize the data and algorithms of the Critical Analysis Reporting Environment (CARE)– a data analysis software package that is developed by the researchers at the University of Alabama (Steil and Parrish, 2009). CARE uses advanced analytical and statistical techniques on the crash and citation data for the State of Alabama to generate valuable information, including hot spot locations, their time and duration, and their severity (in terms of number of fatalities). We utilize this information to manage state trooper resources and patrol routes.

Our work mostly borrows from and contributes in two main areas of operations research: state trooper patrolling models and the orienteering problem. Next, we review and summarize the research on these areas.

### 1.2.1 State Trooper/Plice Patrol Models

The research on police patrols dates back to the early 1970s. The early works are concerned with answering calls for service, mostly related to a police officer servicing a crime call. Hence, mostly queueing models are used (Larson, 1973; Chaiken and Dormont, 1978; Green, 1984; Birge and Pollock, 1989). Other approaches for the patrol routing problem include mathematical modeling (Mitchell, 1972; Wolfler-Calvo and Cordone, 2003; Curtin et al., 2007), heuristic solutions (Wolfler-Calvo and Cordone, 2003; Lauri and Koukam, 2008; Reis et al., 2006), graph theory (Duchenne et al., 2005, 2007; Chawathe, 2007), and simulation (Machado et al., 2003; Santana et al., 2004). Chawathe (2007), as in our paper, considers a road network with hot spots. By means of graph theory, the road network is translated to an edge-weighted graph to find the patrol routes where the weights are related to the importance of the corresponding locations and the topology of the road network. In this paper, the selection of weights is somewhat arbitrary and influences the selection of routes.

One approach for the mathematical modeling of patrol routing problems is to invoke the m-Peripatetic Salesman Problem (m-PSP) that consists of determining $m$ edge disjoint Hamiltonian cycles of minimum total cost on a complete graph. Wolfler-Calvo and Cordone (2003) introduce m-PSP in the design of watchman tours where it is often important to assign a set of edge-disjoint rounds to the watchman in order to avoid repeating the same tour and enhancing security. They solve this model via a decomposition heuristic. Duchenne et al. (2005, 2007) improve the formulation of the m-PSP by defining new polyhedral properties and cuts and describe exact branch-and-cut solution procedures for the undirected m-PSP. The two main differences between this line of work and ours are the time-sensitivity of hot spot coverage and maximization of coverage benefits instead of minimization of travel costs.

Therefore, our model is unprecedented in the patrol routing literature that addresses the design of patrol routes while covering hot spots within their time limits.

### 1.2.2 Orienteering Problem (OP)

The OP is first introduced by Tsiligirides (1984) for the orienteering competition. In this competition, competitors visit as many checkpoints as possible within a time limit where each checkpoint may have different point values depending on difficulty. The competitor with the most points wins the game (Chao et al., 1996a). In a more formal definition, given a weighted graph with profits associated with the vertices, the OP consists of selecting a simple circuit of maximal profit, whose length does not exceed a certain pre-specified bound (Feillet et al., 2005). The OP is also known as the selective traveling salesperson (Laporte and Martello, 1990) or the maximum collection problem (Butt and Cavalier, 1994). The OP arises in many applications including the sport game of orienteering (Chao et al., 1996a), the home fuel delivery problem (Golden et al., 1987), athlete recruiting from high schools (Butt and Cavalier, 1994), routing technicians to service customers (Tang and Miller-Hooks, 2005), and the personalized mobile tourist guide (Vansteenwegen et al., 2009).

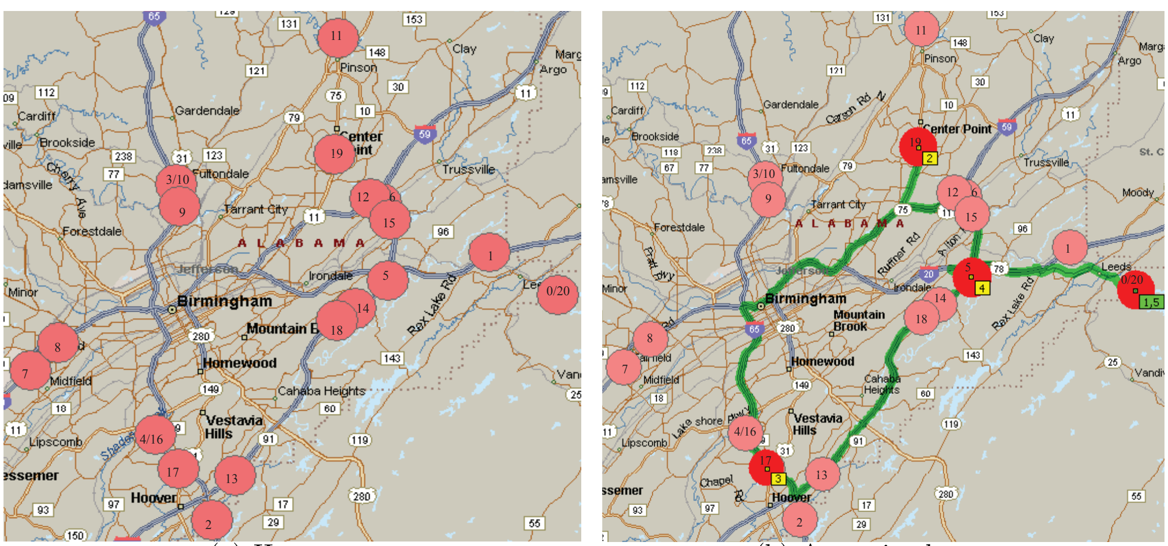
Some important variants of the orienteering problem include the team orienteering problem (TOP)–where a fixed number of paths is considered, the orienteering problem with time windows (OPTW), and the team orienteering problem with time windows (TOPTW). Since Golden et al. (1987) prove that the OP is NP-hard, for OP and its variants only a few researchers resort to exact algorithms. Righini and Salani (2006) and Righini and Salani (2009) use bi-directional dynamic programming, and Boussier et al. (2007) propose an exact branch-and-price approach coupled with a column generation technique. Most other research on OP and the variants have focused on heuristic approaches, including local search (Vansteenwegen et al., 2009), tabu search (Liang et al., 2002; Tang and Miller-Hooks, 2005; Schilde et al., 2009), path relinking (Schilde et al., 2009; Souffriau et al., 2010), ant colony optimization (Liang et al., 2002; Ke et al., 2008; Montemanni and Gambardella, 2009), genetic algorithm (Tasgetiren, 2001), and other metaheuristics (Archetti et al., 2007; Tricoire et al., 2010). A recent review, summarizing all of these variants, solution approaches, and benchmark models, is presented by Vansteenwegen et al. (2010).

As our problem bears similarities to the TOPTW, we discuss the TOPTW literature in more detail. The exact branch-and-price algorithm proposed by Boussier et al. (2007) is generic enough to handle different kinds of OP, including the TOPTW. The different branching rules and acceleration techniques introduced in this paper helps solve problem instances with up to 100 nodes. Montemanni and Gambardella (2009) develop local search and ant colony system algorithms based on the solution of a hierarchic generalization of TOPTW. The algorithms are tested effective for OPTW and TOPTW with up to 288 nodes. Last but not the least, Vansteenwegen et al. (2009) present a straightforward and very fast iterated local search heuristic, which combines an insertion step and a shaking step– reverse insertion operation, to escape from local optima. It performs well on the available data sets, ranging from 3-20 routes and 48-288 nodes. The solution quality is slightly worse than that of Boussier et al. (2007) and Montemanni and Gambardella (2009), but the solution approach requires only a few seconds of computation time, more than 100 times faster.

## 1.3 GENERAL MODEL

Our problem is formally defined as follows. Within a particular county with an established state trooper post and during a particular shift $p$, there are historically established hot spots that are more prone to accidents. These hot spots are defined not only with their location on the mile-posted road network, but also with the time they become "hot." We denote the set of hot spots with $\mathcal{N} = \{1, \ldots, n\}$ where each hot spot $i \in \mathcal{N}$ has an earliest $e_i$ and latest time $l_i$ for its hotness. By definition, $e_i < l_i$. We denote $[e_i, l_i]$ as the time window $TW_i$ of hot spot $i$. Furthermore, we assume that set $\mathcal{N}$ is ordered according to $e_i$ such that $e_1 \leq e_2 \leq \ldots \leq e_n$. We note that the same location can be labeled with two different indices as $i$ and $j$, where $l_i < e_j$ as an indication of two different hot spots. Additionally, we define the dummy nodes 0 and $n + 1$ to denote the start and end of the shift at the state trooper post, respectively. $\mathcal{V} = \{0, n + 1\} \cup \mathcal{N}$ denote the set of all hot spots and the state trooper post. For a certain shift $p$, $e_0 = A_p$ and $l_{n+1} = L_p$, where $A_p$ and $L_p$ are the starting and ending time of the shift $p$. Given the maximum number of state trooper cars $|\mathcal{K}|$ available, we aim to find the best patrol route for each state trooper car $k \in \mathcal{K}$ with critical stops at hot spots to create a deterrence effect.

The problem representation of an example with 19 hot spots is given in Figure 1. In this figure, nodes 0 and 20 represent the state trooper post. Furthermore, hot spot pairs {3, 10} and {4,16} are at the same location. They are marked as separate hot spots due to having two distinct time windows, i.e., they become "hot" twice during the shift. For instance, the location marked with hot spots 4 and 16 becomes "hot" between 7:00-8:30am and 11:00am-12:30pm, respectively. In Figure 1(b), we demonstrate one of the routes of the optimal solution for this example. Even though the state trooper patrol includes hot spots 5, 14, 18, 13, 2, 17, 4, 16, 19, 12, 6, and 15, in that order, only the visits to 5, 17, and 19 fall into their respective time windows, and only these stops count as a deterrent for accidents.



(a) Hot spots (b) An optimal route

Figure 1: A representative example.

Additionally, we let $\mathcal{E} = \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$ define the set of edges. The connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents the underlying road network. We denote the shortest travel time from vertex $i$ to $j$ as $t_{ij} > 0$, $i, j \in \mathcal{V}, i \neq j$. Our objective is to construct the best patrol routes to maximize the total amount of effective service time which falls within $TW_i$ of hot spot $i$, $\forall i \in \mathcal{N}$. For this purpose, we define three sets of decision variables: i) $x_{ijk} = 1$, if state trooper car $k \in \mathcal{K}$ travels from vertex $i$ to $j$, $(i, j) \in \mathcal{E}$, and 0, otherwise. ii) $s_{ik} \geq 0$, the starting time of service for state trooper car $k \in \mathcal{K}$ at vertex $i \in \mathcal{V}$. iii) $f_{ik} \geq 0$, the time state trooper car $k \in \mathcal{K}$ leaves vertex $i \in \mathcal{V}$, i.e., the end of service.

Before proceeding with our model development, we summarize the assumptions of the model:

1. There is a one to one correspondence between a state trooper car and a state trooper, and all of the state trooper cars are identical.

2. One state trooper car is sufficient to cover each hot spot. That is, having multiple state troopers at the same time at a particular location does not augment their deterrence ability.

3. State troopers travel at a constant speed of 60 miles/hour. Therefore, travel time from one hot spot to another is a calculated constant and irrelevant to time of day or day of week.

4. Refueling is possible from any gas station on their patrol route and is not considered.

5. At the beginning of a shift, all state trooper cars start from the same state trooper post 0 and come back to the same location at the end of the shift.

6. A state trooper car is allowed to arrive before $e_i$ and wait until the start time of the hot spot, but its presence is a deterrent only after $e_i$.

7. Since roadway traffic accidents have a weekly pattern, we model the problem for a particular day of the week and shift of the day.

8. Each county is divided into several districts, and each district has only one state trooper division. State troopers are only responsible for their own jurisdiction. We conclude

that each district is independent from each other, thus each district can be solved independently. The formulation below is for a particular district.

Our objective for the Maximum Covering Patrol Routing Problem (MCPRP) is to maximize the total amount of service time, which falls within the time window of a hot spot:

$$\text{Maximize} \sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{K}} (f_{ik} - s_{ik}) \tag{MCPRP}$$

We categorize our constraints under four groups: schedule feasibility, route structuring, visits to hot spots, and integrality and non-negativity constraints.

### Schedule Feasibility Related Constraints

We need to guarantee schedule feasibility with respect to time considerations for each state trooper car $k$, $k \in \mathcal{K}$. If state trooper car $k$ visits vertex $j \in \mathcal{V}$ after a stop at vertex $i \in \mathcal{V}$, i.e., $x_{ijk} = 1$, then the start time at vertex $j$ should be greater than or equal to the finish time of the current vertex $i$ plus the travel time between $i$ and $j$, that is $s_{jk} \geq f_{ik} + t_{ij}$. To ensure schedule feasibility, we need

$$x_{ijk} * (f_{ik} + t_{ij} - s_{jk}) \leq 0,$$

for each $(i, j) \in \mathcal{E}$, and $k \in \mathcal{K}$. We linearize these constraints using a big constant value $M_{ij} = \max\{l_i + t_{ij} - e_j, 0\} \geq 0$ as follows:

$$f_{ik} + t_{ij} - s_{jk} \leq (1 - x_{ijk}) M_{ij}, \qquad \forall k \in \mathcal{K} \text{ and } \forall (i, j) \in \mathcal{E}. \tag{1}$$

Before we proceed with other constraints, we define $\triangle^{+}(i) = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}, e_i + t_{ij} \leq l_j\}$ as the set of vertices that are directly reachable from $i \in \mathcal{V}$ within the time window and $\triangle^{-}(i) = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}, e_j + t_{ij} \leq l_i\}$ as the set of vertices from which $i$ is directly reachable.

Other schedule feasibility constraints include time window restrictions:

$$e_i \sum_{j \in \triangle^+(i)} x_{ijk} \leq s_{ik}, \qquad \forall k \in \mathcal{K} \text{ and } \forall i \in \mathcal{V}. \tag{2}$$

$$l_i \sum_{j \in \triangle^+(i)} x_{ijk} \geq f_{ik}, \qquad \forall k \in \mathcal{K} \text{ and } \forall i \in \mathcal{V}. \tag{3}$$

$$s_{ik} \leq f_{ik}, \qquad \forall k \in \mathcal{K} \text{ and } \forall i \in \mathcal{V}. \tag{4}$$

Constraints (2) establish that the effective start time $s_{ik}$ at vertex $i$ by state trooper car $k$ is at least as large as the earliest time window of vertex $i \in \mathcal{V}$. Constraints (3) state that the end of the effective service time $f_{ik}$ must be less than or equal to the latest time window of vertex $i \in \mathcal{V}$. Finally, Constraints (4) state that the start time of the service by state trooper car $k \in \mathcal{K}$ at vertex $i \in \mathcal{V}$ is less than or equal to the end of the service.

**Route Structuring Constraints**

We characterize the route of a state trooper $k \in \mathcal{K}$ with the following equations:

$$\sum_{j \in \triangle^+(0)} x_{0jk} = 1, \qquad \forall k \in \mathcal{K}. \tag{5}$$

$$\sum_{i \in \triangle^-(j)} x_{ijk} = \sum_{i \in \triangle^+(j)} x_{jik}, \qquad \forall k \in \mathcal{K} \text{ and } \forall j \in \mathcal{N}. \tag{6}$$

$$\sum_{i \in \triangle^-(n+1)} x_{i,n+1,k} = 1, \qquad \forall k \in \mathcal{K}. \tag{7}$$

Constraints (5) ensure all of the state trooper cars leave the state trooper post at the beginning of the shift, and Constraints (7) ensure their return to the post at the end of the shift. Finally, Constraints (6) state the balance at each hot spot, i.e. each state trooper car $k$ that visits hot spot $i$ must leave.

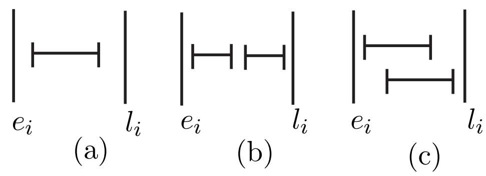



Figure 2: Multiple state troopers at hot spot $i \in \mathcal{N}$

**Constraints Related to Visiting Hot Spots**

It is possible to have multiple cars visiting the same hot spot as in Figures 2(b) and (c). Therefore, we need to account for any potential double counting if there is overlap during the visits of multiple cars, as in Figure 2(c), and eliminate it. The next set of constraints ensure that if multiple coinciding cars are present at the same hot spot at the same time, they contribute to the objective only once. To establish these constraints, we define the following additional decision variables for $i \in \mathcal{V}$ and $k, g \in \mathcal{K}$, $k \neq g$:

$$y_{ik} = \begin{cases} 1, & \text{if state trooper } k \text{ serves vertex } i; \\ 0, & \text{otherwise.} \end{cases} \quad \text{and} \quad u_{ikg} = \begin{cases} 1, & \text{if } s_{ig} \geq f_{ik}; \\ 0, & \text{otherwise.} \end{cases}$$

By definition of $y_{ik}$,

$$\sum_{j \in \triangle^+(i)} x_{ijk} = y_{ik}, \qquad \forall k \in \mathcal{K} \text{ and } \forall i \in \mathcal{N}. \tag{8}$$

$$y_{0,k} = y_{n+1,k} = 1, \qquad \forall k \in \mathcal{K}. \tag{9}$$

Additionally, by definition, $u_{ikg}$ or $u_{igk}$ can only be equal to 1 when both $y_{ik} = 1$ and $y_{ig} = 1$, or else, $u_{ikg} = u_{igk} = 0$ for $i \in \mathcal{V}$. The following constraints establish the relationship between $y_{ik}$ and $u_{ikg}$:

$$u_{ikg} + u_{igk} \leq y_{ik}, \qquad \forall i \in \mathcal{V} \text{ and } k, g \in \mathcal{K}, \ g > k. \tag{10}$$

$$u_{ikg} + u_{igk} \leq y_{ig}, \qquad \forall i \in \mathcal{V} \text{ and } k, g \in \mathcal{K}, \ g > k. \tag{11}$$

$$u_{ikg} + u_{igk} \geq y_{ik} + y_{ig} - 1, \qquad \forall i \in \mathcal{V} \text{ and } k, g \in \mathcal{K}, \ g > k. \tag{12}$$

Now, we are ready to present the constraints that eliminate "double counting" if there are two or more cars at the same time window of a certain vertex. That is, for $i \in \mathcal{V}$, if $y_{ik} = 1$ and $y_{ig} = 1$, then $f_{ik} \leq s_{ig}$ or $s_{ik} \geq f_{ig}$, where $k, g \in \mathcal{K}$ and $k \neq g$:

$$f_{ik} - s_{ig} - M * (1 - u_{ikg}) \leq 0, \qquad \forall i \in \mathcal{V} \text{ and } k,\, g \in \mathcal{K},\ g > k. \tag{13}$$

$$f_{ig} - s_{ik} - M * (1 - u_{igk}) \leq 0, \qquad \forall i \in \mathcal{V} \text{ and } k,\, g \in \mathcal{K},\ g > k. \tag{14}$$

where $M$ is a large constant.

**Integrality and Non-negativity Constraints**

Finally, we state continuous and binary variables:

$$s_{ik}, f_{ik} \geq 0 \quad \text{and} \quad x_{ijk}, y_{ik}, u_{ikg} \in \{0, 1\}, \quad \forall i, j \in \mathcal{V} \text{ and } k,\, g \in \mathcal{K},\ g > k. \tag{15}$$

**Overall Model**

The overall model is to maximize the effective service time for (MCPRP), subject to constraints (1)–(15). We solve this formulation using CPLEX 12.1. However, even for very small instances with 40 hot spots and two state trooper cars, CPLEX runs out of memory.

**Theorem 1** *MCPRP is NP-hard.*

The proof is found in Appendix. Due to Theorem 1, we focus on two two-phase heuristics. These are composed of a construction algorithm and local-search and tabu-search-based improvements. Before we discuss our solution approaches, we note that this model can be used to evaluate other performance measures including "Percentage of Hot Spots Covered (HS%)" and "Percentage of Coverage Length (TW%)".

**HS%:** This performance measure calculates, among all of the hot spots, the percentage covered as a result of the *MCPRP*:

$$HS\% = \frac{\sum_{i\in\mathcal{N}}\sum_{k\in\mathcal{K}} y_{ik} - \sum_{i\in\mathcal{N}}\sum_{g\neq k}(u_{igk}+u_{ikg})}{n} * 100,$$

where the numerator represents the total number of visited hot spots.

**TW%:** This performance measure calculates the percentage of total available time serviced by the *MCPRP*:

$$TW\% = \frac{\sum_{i\in\mathcal{N}}\sum_{k\in\mathcal{K}}(f_{ik}-s_{ik})}{\sum_{i\in\mathcal{N}}(l_i-e_i)} * 100.$$

In this measure, the numerator is the service time returned by the *MCPRP*, and the denominator is the total time window length.

## 1.4 SOLUTION APPROACHES

Our solution approaches build on the following characterization of the optimal solution.

**Proposition 1** *If the optimal sequences of covered hot spots are known, in the optimal solution, for each state trooper $k \in \mathcal{K}$, for a visited hot spot $i \in \mathcal{N}$*

$$f_{ik} = \begin{cases} \min\{l_i,\ T - t_{i,n+1}\}, & \textit{if } i \textit{ is the last hot spot visited on the route of } k; \\ l_i, & \textit{otherwise}; \end{cases}$$

*where $T = L_p$, the end of the shift $p$.*

This proposition states that, in the optimal solution, the end of service time at a visited hot spot $i$ depends on the order of $i$ in the route. If hot spot $i \in \mathcal{N}$ is the last hot spot on route $k$, $f_{ik}$ is the minimum of the latest time window of hot spot $i$ and $T - t_{i,n+1}$ (time that is required to get back to the post within the shift duration). Otherwise, hot spot $i \in \mathcal{N}$ is an intermittent node in the route and $f_{ik} = l_i$. In other words, state trooper $k$ can stay until the latest time window of each hot spot that is on the route. The complete proof is presented in Appendix.

This proposition states that if there is excess time in a route–the times other than the effective coverage and travel among hot spots, whether a state trooper is going to wait after covering a certain hot spot or before the time window of the next hot spot in the route does not make a difference in construction of the routes or the objective value. Therefore, by this proposition, we arbitrarily place any excess time at the beginning of the next hot spot without loss of generality. These characteristics are due to two assumptions of the problem: i) the travel time $t_{ij}$ is fixed, as travel speed is constant of 60 miles/hour; and ii) all hot spots have the same priority. If either one of these assumptions is relaxed, then the excess time may not be arbitrarily placed in a route as it influences the order of nodes covered, travel times, and coverage and hence impacts the optimal solution. We report results related to relaxing the hot spots priorities in the computational experiments section.

### 1.4.1 Construction Algorithm

Based on Proposition 1, we develop a construction algorithm with two parts involving route initialization and hot spot insertion.

#### 1.4.1.1 Route Initialization Algorithm

First, we define the following two algorithm parameters $H_{limit}$ and $T_{limit}$ that help us in building the initial routes:

- $H_{limit}$ provides an upper bound on the number of hot spots to be considered for insertion into a route. Our hot spots are ordered according to the start time of their time windows. To avoid big time gaps between the start times of two consecutive hot spots, and hence, to eliminate any potential excess waiting, after a node is inserted into a route, we only consider the next $H_{limit}$ hot spots as the potential next hot spot to be included in this route. We set $H_{limit}$ as $\lceil n/|\mathcal{K}| \rceil$.

- $T_{limit}$ is a clustering factor where travel time from one hot spot to the next hot spot cannot exceed a certain time span. After preliminary experimentations, we set $T_{limit}$ to 100 minutes, which is reasonable given that for the instances we tested $T = 480$ minutes. If the travel time from a currently visited hot spot to the next one exceeds 100 minutes, then the algorithm is not going to consider that point.

Hence, $H_{limit}$ provides a temporal limit while $T_{limit}$ provides a spatial restraint on the initial routes.

**Algorithm 1** Procedure *RouteInitilization*.

1: Uncovered hot spot set $\mathcal{U} \leftarrow \mathcal{N}$. For $k \in \mathcal{K}$, initialize $Route_k \leftarrow \emptyset$.
2: **for** $\forall k \in \mathcal{K}$ **do**
3: $\quad Route_k \leftarrow Route_k \cup \{0\}$.
4: $\quad i^* \leftarrow \arg\max_{i \in \mathcal{U}}\{l_i - \max(e_i, t_{0i}) : i \leq H_{limit}, t_{0i} \leq T_{limit}, t_{0i} \leq l_i\}$.
5: $\quad s_{i^*,k} \leftarrow \max\{e_{i^*}, t_{0,i^*}\}$ and $f_{i^*,k} \leftarrow l_{i^*}$. $Route_k \leftarrow Route_k \cup \{i^*\}$. $\mathcal{U} \leftarrow \mathcal{U} \setminus \{i^*\}$.
6: **end for**
7: **for** $\forall k \in \mathcal{K}$ **do**
8: $\quad i \leftarrow Route_k.lastHotSpot$.
9: $\quad$ **for** $\forall j \in \mathcal{U}$ such that $i < j \leq (i + H_{limit})$, $t_{ij} \leq T_{limit}$, and $l_i + t_{ij} < l_j$ **do**
10: $\quad\quad$ **if** $l_j + t_{j,n+1} < T$ **then**
11: $\quad\quad\quad i^* \leftarrow \arg\max_{j \in \mathcal{U}}\{l_j - \max(e_j, l_i + t_{ij})\}$.
12: $\quad\quad\quad s_{i^*,k} \leftarrow \max\{e_{i^*}, l_i + t_{i,i^*}\}$ and $f_{i^*,k} \leftarrow l_{i^*}$. $Route_k \leftarrow Route_k \cup \{i^*\}$.
13: $\quad\quad\quad$ **if** $e_{i^*} < l_i + t_{i,i^*}$ **then**
14: $\quad\quad\quad\quad l_{i^*} \leftarrow l_i + t_{i,i^*}$.
15: $\quad\quad\quad$ **else**
16: $\quad\quad\quad\quad \mathcal{U} \leftarrow \mathcal{U} \setminus \{i^*\}$.
17: $\quad\quad\quad$ **end if**
18: $\quad\quad$ **else**
19: $\quad\quad\quad$ **if** $l_i + t_{ij} < T - t_{j,n+1}$ **then**
20: $\quad\quad\quad\quad i^* \leftarrow \arg\max_{j \in \mathcal{U}}\{T - t_{j,n+1} - \max(e_j, l_i + t_{ij})\}$;
21: $\quad\quad\quad\quad s_{i^*,k} \leftarrow \max\{e_{i^*}, l_i + t_{i,i^*}\}$ and $f_{i^*,k} \leftarrow T - t_{i^*,n+1}$. $Route_k \leftarrow Route_k \cup \{i^*\}$.
22: $\quad\quad\quad\quad$ Repeat Steps 13 to 17.
23: $\quad\quad\quad$ **end if**
24: $\quad\quad$ **end if**
25: $\quad$ **end for**
26: **end for**

The *RouteInitialization* heuristic, where the pseudo-code is given in Algorithm 1, builds on a greedy principle. Each state trooper car starts from the state trooper post at the beginning of the shift. Among all of the hot spots, within the distance range $T_{limit}$ and time range $H_{limit}$, if the arrival time of state trooper $k$ from hot spot $i$ at one of these hot spots, say hot spot $j$, is before the end of the time window ($l_i + t_{ij} < l_j$), the heuristic picks the hot spot that contributes to the objective the most as the next place to visit, that is $i^*$. The maximum contribution is calculated as $\max_j\{l_j - \max(e_j, l_i + t_{ij})\}$. Then, the start and finish times of service at $i^*$ are calculated by comparisons between the arrival time at $i^*$ and earliest time windows, and latest time windows respectively, as in line 12. After the next hot spot is selected, the algorithm is divided into two cases as described in steps 10 and 19: whether or not there is enough time for the state trooper to *fully* service the next hot spot and be back at the state trooper post before the end of the shift. In the first case, there exist hot spots where the coverage and travel-to-post times are within the shift duration. Among these hot spots, the hot spot $i^*$ with the maximum coverage potential is added to the route. Steps 13 through 17 check for potential multi-car visits. Specifically, if a state trooper arrives before or at $e_{i^*}$, the hot spot $i^*$ is covered fully from $[e_{i^*}, l_{i^*}]$ and is removed from $\mathcal{U}$. Otherwise, hot spot $i^*$ is split into uncovered $[e_{i^*}, s_{i^*,k}]$ and covered $[s_{i^*,k}, f_{i^*,k}]$ parts. In this situation, $i^*$ with an updated $l_{i^*}$ stays in $\mathcal{U}$. For the second case, starting with Step 19, it is not feasible for a state trooper to stay until the end of the time window of hot spot $j$ due to approaching the end of the shift. Therefore, by factoring in the travel time from hot spot $j$ to the state trooper post $n+1$, the state trooper can stay until $T - t_{j,n+1}$. Among all of the partially coverable hot spots, the one with the maximum coverage gain $i^*$ is selected. Again, to ensure multi-car visits, steps 13 through 17 are repeated. In this way, initial $|\mathcal{K}|$ routes are created in parallel.

#### 1.4.1.2 Insertion Algorithm

After route initialization, to cover the hot spots which are not covered yet, we proceed with the following insertion algorithm. To insert an uncovered hot spot $\bar{i} \in \mathcal{U}$ before a hot spot $i$ in a certain route $k \in \mathcal{K}$, we first check the time window feasibility of hot spot $i$, i.e., arrival time at hot spot $i$ is less than the latest time window of the hot spot, that is, $l_{\bar{i}} + t_{\bar{i},i} < l_i$. In this algorithm, starting with the first hot spot of the first route we check if we can insert any more hot spots until it is not feasible. The search ends when all of the $|\mathcal{K}|$ routes are checked.

If it is feasible (in terms of travel and coverage times) to insert a new hot spot $\bar{i}$ *right before* hot spot $i$ on route $k$, this insertion will not influence the start or finish times of hot spots on this route prior to hot spot $i-1$. Insertion of $\bar{i}$ will only shift the starting time of the hot spot $i$, $s_{ik}$, to $s_{i'k}$. Hot spots after $i$ will not be affected since the finishing time at $i$

remains unchanged, i.e., $f_{ik} = l_i$. The additional coverage of hot spot $\bar{i}$ benefits the objective function by as much as $f_{\bar{i},k} - s_{\bar{i},k}$, where $f_{\bar{i},k} = l_{\bar{i}}$ and $s_{\bar{i},k} = \max(e_{\bar{i}}, l_{i-1} + t_{i-1,\bar{i}})$. On the other hand, the coverage of hot spot $i$ may potentially be reduced due to the late start $s'_{i,k}$ at hot spot $i$. The change in the objective due to insertion of $\bar{i}$ *right before* hot spot $i$ is given as:

$$\begin{aligned} \delta &= \text{Benefit After } \bar{i} \text{ Insertion} - \text{Original Benefit} \\ &= \{f_{\bar{i},k} - s_{\bar{i},k}\} + \{f_{ik} - s'_{ik}\} - \{f_{ik} - s_{ik}\} \\ &= l_{\bar{i}} - s_{\bar{i},k} - (s'_{i,k} - s_{i,k}). \end{aligned} \tag{16}$$

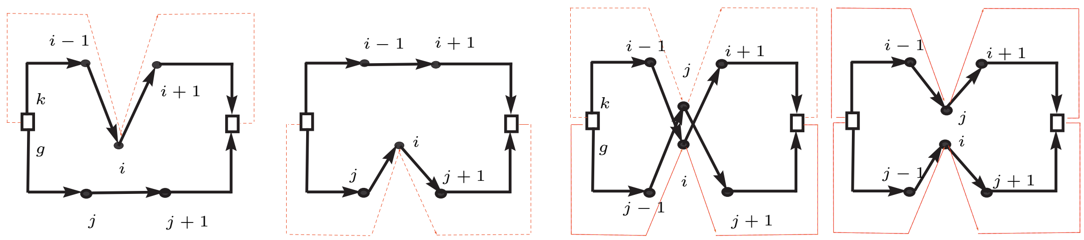
When $\delta > 0$, there is value in including $\bar{i}$ between hot spots $i-1$ and $i$; or otherwise, we continue to check the next uncovered hot spot.

### 1.4.2 Improvement Algorithms

As mentioned above, hot spots are inserted sequentially. The construction algorithm is affected by the selection and order of the subsequently inserted hot spots. The improvement algorithms address this issue by utilizing modified versions of relocate and exchange operators introduced originally for the vehicle routing problem with time windows (Bräysy and Gendreau, 2005; Braysy and Gendreau, 2005b). The relocate operator finds improvements by moving one hot spot from one route to another route whereas exchange operator exchanges hot spots between two different routes. The modification step involves revoking the insertion algorithm after each move.

#### 1.4.2.1 Relocate Operator

In Figure 8(a), we present the relocate operator, where hot spot $i$ from the origin route $k$ is moved into the destination route $g$, $k \neq g$. In the figure, we also represent the other routes visiting $i$– due to the possible visits by multiple cars, in red dotted lines. We let $(s_{ik}, f_{ik})$ and $(s_{ig}, f_{ig})$ as well as $(s_{i+1,k}, f_{i+1,k})$ and $(s'_{i+1,k}, f'_{i+1,k})$ denote the start and finish times at




(a) Relocate Operator (b) Exchange Operator

Figure 3: Neighborhood search operators.

hot spots $i$ and $i+1$ before and after the move, respectively. Hot spots $j$ and $j+1$ follow a similar notation. After the move, the change in the objective is

$$\begin{aligned}\Delta &= (f_{ig} - s_{ig}) - (f_{ik} - s_{ik}) + (f'_{i+1,k} - s'_{i+1,k}) - (f_{i+1,k} - s_{i+1,k}) + (f'_{j+1,g} - s'_{j+1,g}) - (f_{j+1,g} - s_{j+1,g}) \\ &= (s_{ik} - s_{ig}) + (s_{i+1,k} - s'_{i+1,k}) + (s_{j+1,g} - s'_{j+1,g}),\end{aligned}$$

as finishing times before and after the move are the same. However, modification of the start times of the coverage is more complicated due to the possibility of covering a hot spot with multiple cars. If hot spot $i$ is only visited by route $k$ or $k$ is the first of multiple visits to hot spot $i$, the start time after the move is obtained by comparing the arrival time at hot spot $i$ from a visit at $j$ with the earliest time window hot spot $i$, i.e., $s_{ig} = \max\{f_{jg} + t_{ij}, e_i\}$. Otherwise, hot spot $i$ is visited by multiple cars and route/car $k$ is an intermittent car. That is, the hot spot $i$ is covered by some other car(s) until $s_{ik}$. Therefore, the start time after the move is obtained by comparing the arrival time at hot spot $i$ from $j$ and $s_{ik}$, i.e., $s_{ig} = \max\{f_{jg} + t_{ij}, s_{ik}\}$. A similar check takes place for updating $s'_{i+1,k}$ and $s'_{j+1,g}$.

If $\Delta \leq 0$, the relocate operator is not successful in generating a better solution and is not pursued any further. We move onto the next route and/or hot spot. Otherwise, i.e., $\Delta > 0$, we invoke the insertion algorithm again as relocation may open up additional possibilities to insert an uncovered hot spot. We check if an uncovered hot spot can be inserted between the nodes defined by the modified arcs one by one: $(i-1, i+1)$, $(j, i)$, and $(i, j+1)$. We let $\delta_1$, $\delta_2$, $\delta_3$ be the benefits of inserting an uncovered hot spot before $i+1$, $i$, and $j+1$, respectively.

Each one of these benefits is calculated as in Equation 16. If $\delta_1 > 0$, the insertion before $i+1$ is accepted and updated benefit $\hat{\Delta}$ is set as $\Delta + \delta_1$. Otherwise, if $\delta_2 > 0$, the insertion before $i$ is accepted and $\hat{\Delta}$ is set as $\Delta + \delta_2$. Finally, if $\delta_3 > 0$, $\hat{\Delta}$ is set as $\Delta + \delta_3$. If none of the insertions are favorable, i.e. $\delta_a < 0$ for $a = 1, 2, 3$, the $\hat{\Delta}$ is the same as $\Delta$. Among all of the positive $\hat{\Delta}$ obtained through the whole relocate neighborhood, we pick the one that provides the maximum benefit and implement the relocate (and, if there is one, insertion) associated with that maximum $\hat{\Delta}$. That is, we use the Global Best (GB) acceptance rule.

#### 1.4.2.2 Exchange Operator

In Figure 8(b), we present the exchange operator where two hot spots $i$ and $j$ swap routes simultaneously. As in Figure 8(a), the dotted red lines represent the possibility of other state trooper car(s) covering hot spots $i$ and $j$. After the swap, the start times of the hot spots $i$, $i+1$, $j$, and $j+1$ will be modified. The corresponding change in the objective is

$$\begin{aligned}\Delta &= (f_{ig} - s_{ig}) - (f_{ik} - s_{ik}) + (f'_{i+1,k} - s'_{i+1,k}) - (f_{i+1,k} - s_{i+1,k}) \\ &\quad +(f_{jk} - s_{jk}) - (f_{jg} - s_{jg}) + (f'_{j+1,g} - s'_{j+1,g}) - (f_{j+1,g} - s_{j+1,g}) \\ &= (s_{ik} - s_{ig}) + (s_{jg} - s_{jk}) + (s_{i+1,k} - s'_{i+1,k}) + (s_{j+1,g} - s'_{j+1,g}).\end{aligned}$$

Similar to the relocate operator, these start times are influenced by the number of state trooper cars visiting the hot spot and the order of the cars. In particular,

$$s_{ig} = \begin{cases} \max\{f_{j-1,g} + t_{j-1,i},\ e_i\}, & k \text{ is the 1st visit;} \\ \max\{f_{j-1,g} + t_{j-1,i},\ s_{ik}\}, & \text{O/W.} \end{cases} \qquad s_{jk} = \begin{cases} \max\{f_{i-1,k} + t_{i-1,j},\ e_j\}, & g \text{ is the 1st visit} \\ \max\{f_{i-1,k} + t_{i-1,j},\ s_{jg}\}, & \text{O/W.} \end{cases}$$

The start times $s'_{i+1,k}$ and $s'_{j+1,g}$ are calculated in a similar manner.

If $\Delta > 0$, the exchange is a candidate to be accepted. As in relocate operator, the exchange may provide a possibility to insert an uncovered hot spot between $(i-1, j)$, $(j, i+1)$, $(j-1, i)$, and $(i, j+1)$. The benefits of insertion on these arcs are calculated as $\delta_1$, $\delta_2$, $\delta_3$, and $\delta_4$, respectively, as in Equation 16. The insertion is evaluated in that order, and the first insertion with a positive benefit, i.e., $\delta_a > 0$ for $a = 1, 2, 3, 4$, is accepted. The total benefit $\hat{\Delta}$ is updated as $\Delta + \delta_a$. If none of the insertions return a benefit, then $\hat{\Delta}$ is just set to $\Delta$. Similar to relocate operator, the exchange operator is implemented using the GB criteria. The exchange (and potential insertion) associated with the largest $\hat{\Delta}$ in the neighborhood is accepted. After the exchange (and the potential insertion), the routes and the uncovered hot spot set $\mathcal{U}$ are updated accordingly.

#### 1.4.2.3 Local Search

After introducing all of the neighborhood search components, Figure 4 depicts how these play a role in our local search implementation. In the first stage of improvement, the algorithm keeps looping through the relocate operator embedded with insertion step, until no improvement is found. Note that after the relocate operator embedded with insertion step, the insertion algorithm is called again because if there is any move, the $\mathcal{U}$ set and routes are updated. Thus, there is a chance to insert an uncovered hot spot into any of the existing routes. In the third stage of improvement, the exchange operator embedded with insertion step keeps searching until no further improvement can be found, followed by the insertion step for the same reason as the first stage of improvement. The local search terminates when no further improvement is available.

#### 1.4.2.4 Tabu Search

Based on the fact that local search can be trapped at a local optimum, we also apply a tabu search algorithm as a part of the improvement step.

In our implementation, tabu list consists of two attributes: state trooper car index and hot spot's identification. Specifically, if the most recent solution includes covering hot spot $i$ by state trooper $k$, then the $(i, k)$-pair is marked as tabu. The tabu list length and tabu tenure are set to $5 \times \lfloor \sqrt{n} \rfloor$, directly correlated with the total number of hot spots $n$. In the

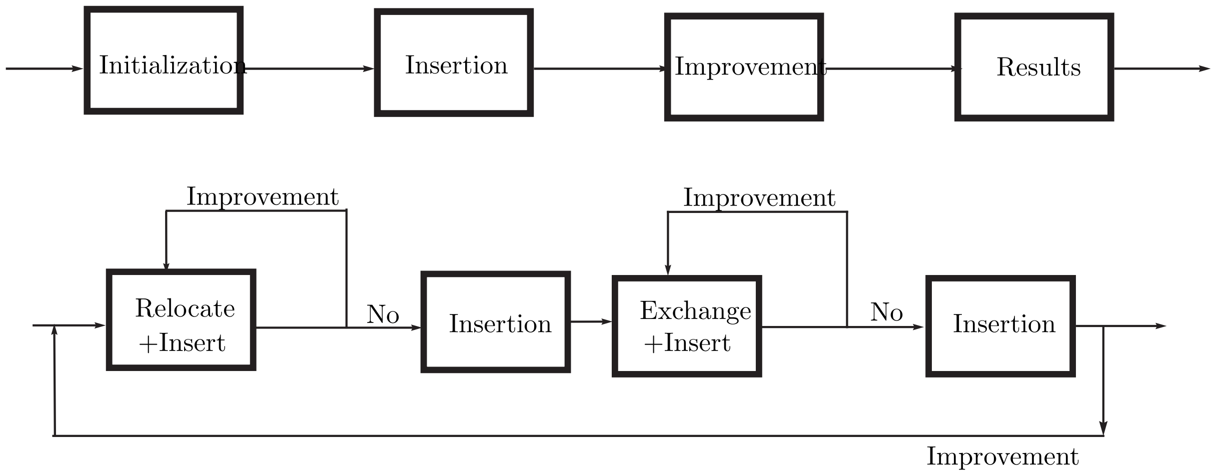



Figure 4: Local search and improvement flow charts.

neighborhood, the relocate operator is followed by the exchange operator. Each operation is conducted over all of the routes and visited hot spots. Random numbers determine the starting hot spot and the starting route number for each operator. Once the search starts, it sweeps through all of the hot spots and routes exhaustively. If it is feasible to carry out a particular operation, both state trooper car and visited hot spot indices are added into the tabu list. With the relocate operator, only the relocated hot spot and its corresponding state trooper indices are added into the tabu list. On the other hand, with the exchange operator, both of the exchanged hot spots and their corresponding route indices are added into the tabu list. As an aspiration criteria, tabu is only overridden when the newly obtained objective is better than the best one found thus far.

## 1.5 COMPUTATIONAL EXPERIMENTS

### 1.5.1 Performance Based Experiments

In order to test the performance and effectiveness of the model and heuristic approaches, we conduct a series of numerical studies on randomly generated problems ranging from small to moderately large sized ones as well as on real life data captured from CARE (see Section 1.2).

In order to benchmark the quality and runtime of our heuristics, we also run CPLEX 12.1 for all of the instances. We implement and run the algorithms using C++ on a Dell Poweredge 6850 with four dual-core 3.66GHz Xeon processors and 8GB of memory.

#### 1.5.1.1 Experiment with Randomly Generated Data

We randomly pick 10, 20, and 40 locations on the highway as well as their corresponding earliest and latest time windows from a pool of real life data, with 20 instances in each data set. Both of these algorithms are tested when there are up to 8 state trooper cars available, i.e., a total of 480 ($3 \times 20 \times 8$) instances.

We compare the solutions returned by local search (LS) and tabu search (TS) with the ones obtained from CPLEX as shown in Table 1. Unfortunately, CPLEX runs out of memory for even relatively small instances, such as the case when 2 state trooper cars are available for 40 hot spots. We evaluate our heuristics by calculating the percentage of gap between objective returned by our heuristics and lower bound (LB) of CPLEX, which is defined as $\textbf{Gap} = \frac{Objective - LB}{LB} * 100$. Note that since we have a maximization problem, the lower bound returns the best feasible solution that CPLEX can obtain and a positive gap indicates that the heuristics outperform the best feasible solution returned by CPLEX. In Table 1, we report both average (Avg.) and maximum (Max.) gap that demonstrate the best performance of the heuristics. We also report number of times that CPLEX is able to find optimal solution out of all 20 instances, contained in the column of "No. opt." and number of times that LS/TS is at least as good as LB returned by CPLEX, contained in the column of "No. best".

Table 1: Performances of LS and TS for random data.

| Data Set | No. Cars | No. Instances | CPLEX No. opt. | LS Avg. | LS Max. | LS No. best | TS Avg. | TS Max. | TS No. best |
|---|---|---|---|---|---|---|---|---|---|
| 10HS | 3 | 20 | 20 | -1.4 | 0.0 | 18 | -1.4 | 0.0 | 18 |
| | 4 | 20 | 3 | 0.0 | 0.0 | 20 | -2.1 | 0.0 | 19 |
| | 5 | 20 | 1 | 0.1 | 3.4 | 20 | 0.1 | 3.4 | 20 |
| | 6 | 20 | 1 | 0.1 | 3.4 | 20 | 0.1 | 3.4 | 20 |
| | 7 | 20 | 0 | 0.1 | 3.4 | 20 | 0.1 | 3.4 | 20 |
| | 8 | 20 | 0 | 0.1 | 3.4 | 20 | 0.1 | 3.4 | 20 |
| 20HS | 3 | 20 | 2 | -1.3 | 0.0 | 5 | -1.5 | 0.0 | 4 |
| | 4 | 20 | 2 | -1.0 | 0.0 | 8 | -1.0 | 0.0 | 5 |
| | 5 | 20 | 2 | -0.8 | 0.0 | 12 | -0.9 | 0.0 | 15 |
| | 6 | 20 | 0 | -0.3 | 0.0 | 16 | -0.8 | 0.0 | 16 |
| | 7 | 20 | 0 | -0.5 | 0.0 | 17 | -0.5 | 0.0 | 17 |
| | 8 | 20 | 0 | -0.1 | 0.0 | 17 | -0.1 | 0.0 | 17 |
| 40HS | 3 | 20 | 0 | -4.9 | 0.0 | 0 | -5.7 | 0.0 | 0 |
| | 4 | 20 | 0 | -2.6 | 0.0 | 0 | -3.3 | 0.0 | 0 |
| | 5 | 20 | 0 | -0.5 | 4.1 | 4 | -1.3 | 1.9 | 2 |
| | 6 | 20 | 0 | -0.9 | 1.4 | 8 | -1.3 | 1.1 | 3 |
| | 7 | 20 | 0 | 0.0 | 4.4 | 12 | -0.4 | 4.4 | 8 |
| | 8 | 20 | 0 | 0.1 | 3.3 | 14 | -0.1 | 2.6 | 12 |

In Table 1, we observe that CPLEX has a deteriorating performance as the number of hot spots and state trooper cars increase. On the contrary, for these instances where CPLEX is struggling, the number of times finding at least as good as LB of CPLEX ("No. best") is increasing for our heuristics. Specifically, our heuristics are able to find at least as good solution as LB of CPLEX for 10 HS case and 20 HS case most of the time and find some good solutions for 40 HS case, especially with higher number of cars. In fact, the heuristics return slightly better solutions when there are a higher number of hot spots and state trooper cars. With respect to the performance comparison between LS and TS, even though there is not much gap difference for LS and TS, LS still performs slightly better than TS especially for higher number of hot spots.

From the perspective of runtime of local search or tabu-based improvement, both are less than 15 seconds even for instances with 40 hot spots. On the contrary, the more cars there are and the bigger the road network is, the longer it takes CPLEX to find an optimal solution. For instance, it typically takes around $1-2$ hours for CPLEX to find an optimal solution (for smaller instances) or just a LB (for larger instances). Thus, we conclude that our heuristic approaches are more practical since state troopers need to respond to road condition changes relatively frequently.

#### 1.5.1.2 Experiment with Real Life Data

We also solve the real life instances obtained from the CARE database and optimize covering and routing for state troopers on the highways by work shift, by day of week, and by region. Due to the large number of tests, we select three representative areas with a large number of hot spots: Jefferson County rural area (Jeff), the Mobile area (Mob), and Tuscaloosa County rural area (Tus). The most representative days and times for the experiment are Monday, Friday, and Saturday with three shifts: morning shift from 7:00am to 3:00pm, afternoon shift from 3:00pm to 11:00pm, and evening shift from 11:00pm to 7:00am. As the other weekdays (Tuesday through Thursday) mimic Monday and Sunday mimics Saturday, we do not report the results for these days.

In Table 2, we present the results for local and tabu search, respectively. Note that the data instances are referred with the first letter representing the day (Monday) and the second letter as that of the work shift. For instance, MM refers to the Monday morning shift. With three work days and three shifts, there are a total of nine instances in every county. Each instance is tested with varying state trooper cars from 3 to 8. At the last row of each county, we summarize the number of optimal solutions CPLEX returned. For each instance with a particular number of state troopers, we report the gap between objective returned by local and tabu search and LB of CPLEX, respectively. A positive gap refers to a better objective value by our heuristics, whereas a negative gap indicates that the best feasible solution returned by the CPLEX is better.

Table 2: Performances of LS and TS for real data.

| Instances | | LS | | | | | | TS | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **3** | **4** | **5** | **6** | **7** | **8** | **3** | **4** | **5** | **6** | **7** | **8** |
| Jeff | MM | -1.5 | -7.1 | 0.0 | 0.0 | 0.0 | 0.6 | -1.5 | -7.1 | 0.0 | 0.0 | 0.0 | 0.6 |
| | MA | -5.8 | -6.1 | -7.4 | -0.2 | -2.8 | -1.0 | -7.6 | -7.4 | -8.6 | -0.5 | -2.3 | -0.3 |
| | ME | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | FM | -2.6 | -2.9 | 0.0 | 0.0 | 0.0 | 0.0 | -2.6 | -2.9 | 0.0 | 0.0 | 0.0 | 0.0 |
| | FA | -1.2 | -2.3 | 0.1 | 0.0 | 0.0 | -0.6 | -1.2 | -2.3 | -2.1 | 0.0 | 0.0 | -0.6 |
| | FE | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | SM | 0.0 | 3.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3.2 | 0.0 | 0.0 | 0.0 | 0.0 |
| | SA | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | SE | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | No. CPX Opt. | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Mob | MM | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | MA | -3.2 | -2.8 | 0.0 | -2.5 | 0.0 | 0.0 | -3.2 | -2.8 | 0.0 | -0.3 | 0.0 | 0.0 |
| | ME | -1.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -1.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | FM | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | FA | -1.7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -1.7 | -0.8 | 0.0 | 0.0 | 0.0 | 0.0 |
| | FE | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -4.1 | 0.0 | 0.0 | 0.0 | 0.0 |
| | SM | -4.7 | -0.2 | -0.7 | 0.0 | 0.0 | 0.0 | -4.7 | -0.2 | -0.3 | 0.0 | 0.0 | 0.0 |
| | SA | 6.6 | 19.5 | 0.0 | 0.0 | 0.0 | 0.0 | 6.6 | 19.5 | 0.0 | 0.0 | 0.0 | 0.0 |
| | SE | -1.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -1.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | No. CPX Opt. | 5 | 2 | 0 | 0 | 0 | 0 | 5 | 2 | 0 | 0 | 0 | 0 |
| Tus | MM | -0.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -0.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | MA | -0.2 | -0.8 | 0.0 | 0.5 | 0.7 | -0.3 | -0.2 | -0.8 | 0.0 | 0.5 | 0.7 | -0.3 |
| | ME | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | FM | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | FA | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -2.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | FE | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | SM | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -4.6 | 0.0 | 0.0 | 0.0 | 0.0 |
| | SA | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | SE | -1.4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -1.4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | No. CPX Opt. | 5 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 |

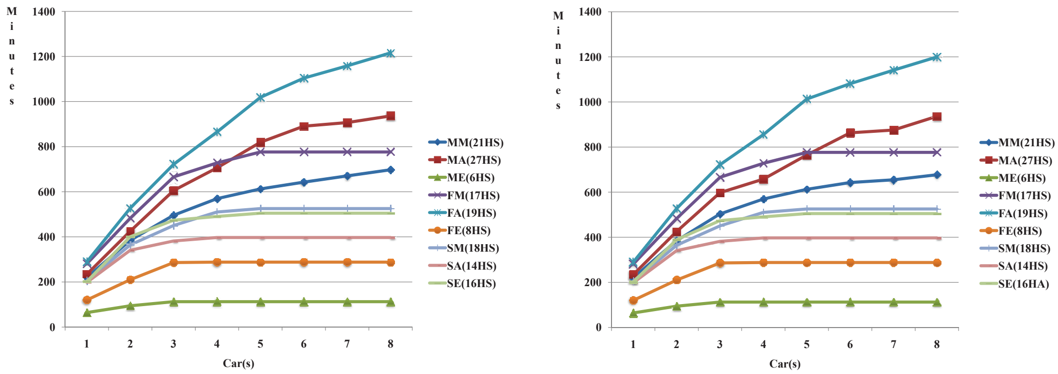
Most of the time, the gap between the heuristics and CPLEX is nonnegative since the solution quality is as good as or better than that of LB of CPLEX. Most gaps fall into a range between $-1\%$ and $1\%$, with very few outliers. Some of these extremes are the negative gaps of $-5.8\%$, $-6.1\%$, and $-7.4\%$ for Jefferson during Monday afternoon shift with three, four, and five state trooper cars, respectively. In this particular instance, the number of hot spots is 27 with varying durations. With limited number of state trooper cars and excess amount of hot spots to cover, the heuristics tend to not perform as good since, in general, they do depend on the improvements (relocate or exchange) among a number of routes.

On the other extreme, there is a positive gap of 19.5% for Mobile during Saturday afternoon shift with four state trooper cars. This is attributed to the poor performance of CPLEX. However, this is not due to our formulation or the gap. More specifically, for this instance as well as the instances marked in blue, CPLEX claims to reach the optimum with the lower bound equal to the upper bound. However, our heuristics *return a better solution* than the claimed CPLEX optimum. After double checking into these solutions with manual calculations, we observe that the solutions returned by the heuristics are indeed feasible and optimal. We reported our model and these problematic instances to ILOG technical support group. They confirmed that there is an internal failure in the CPLEX engine while solving these instances. Now, these instances are added to their test bed to improve the CPLEX engine.

In summary, as the problem size grows, CPLEX has a harder time in obtaining reasonable solutions. In comparison between LS and TS, LS outperforms TS slightly most times. Again, for the computational time, our heuristics provide results within seconds; while CPLEX takes at least couple of hours to find a relatively good feasible solution.

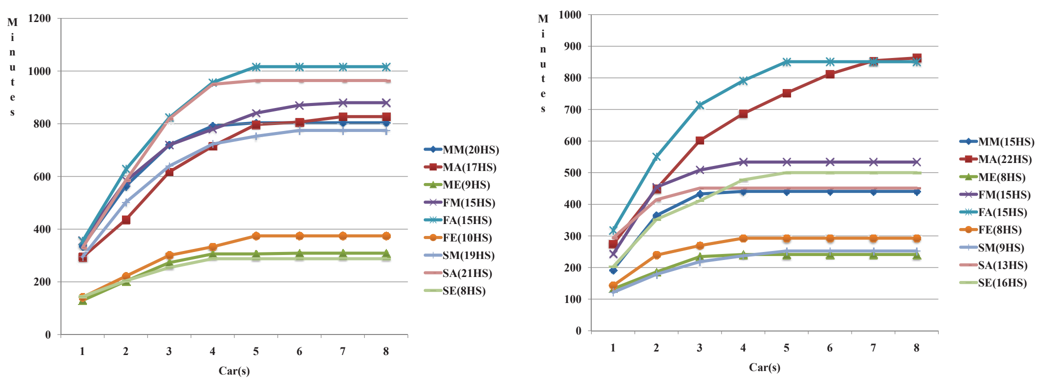
### 1.5.2 Managerial Insights

In this section, we provide managerial insights for decision makers based on our solutions with real data. In Figures 5 and 6, we plot the objective value of *MCPRP* returned by LS and TS with respect to different state trooper cars, respectively. From the plotted charts, we can determine how many state trooper cars are needed for each data set. Intuitively, as the number of state troopers on patrol increases, hot spot coverage improves. However, there are diminishing returns with the addition of each state trooper. One interesting observation is that, as there are more hot spots, the objective is higher. This is due to higher potential coverage. However, in Jefferson County, the top line corresponds to Friday Afternoon with 19 hot spots. This particular instance returned a higher objective compared to, say, Monday Afternoon with 27 hot spots. Investigating this phenomenon further, we found out that the time window of hot spots are not equal. In the data set with 19 hot spots, most of the hot




(a) Local Search (b) Tabu Search

Figure 5: The coverage with LS and TS due to different state trooper cars in Jefferson county.




(a) Mobile Local Search (b) Tuscaloosa Local Search

Figure 6: The coverage with LS in the city of Mobile and Tuscaloosa county.

spots are "hot" for more than an hour, whereas in the data set with 27 hot spots, most of the hot spots are only "hot" for half an hour. Hence, the objective not only depends on the total number of hot spots available, but also length of each hot spot.

Investigating Figures 5 and 6, we can help identify how many state troopers are needed in each shift on each day. For instance, for Jefferson County, for Monday and Friday evenings, three state trooper cars suffice. However, for Saturday evening, at least five cars are needed. Furthermore, for Monday and Friday afternoons, even eight cars may not be enough. This analysis not only provides a good basis for how to allocate resources, but it also demonstrates how the adverse effects of lack of resources (i.e., potential budget and personnel cuts) can be alleviated.

Note, theoretically speaking, all lines should be concave, however in part ($b$) of Figure 5, the objective of Monday afternoon is not concave, since they are returned by our heuristics.

Table 3: Service measure performances by incremental state troopers.

| Data Set | | MM | MA | ME | FM | FA | FE | SM | SA | SE |
|---|---|---|---|---|---|---|---|---|---|---|
| Jeff | Cars | 8 | 8 | 3 | 5 | 8 | 3 | 5 | 4 | 5 |
| | HS | 21 | 27 | 6 | 17 | 19 | 8 | 18 | 14 | 16 |
| | HS% | 90 | 93 | 67 | 100 | 100 | 100 | 89 | 100 | 100 |
| | TW | 810 | 1110 | 179 | 960 | 1410 | 299 | 600 | 449 | 570 |
| | TW% | 86 | 84 | 63 | 81 | 86 | 96 | 88 | 88 | 89 |
| Mob | Cars | 5 | 7 | 4 | 6 | 5 | 5 | 6 | 5 | 4 |
| | HS | 20 | 17 | 9 | 15 | 15 | 10 | 19 | 21 | 8 |
| | HS% | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | TW | 840 | 870 | 330 | 930 | 1050 | 420 | 910 | 1020 | 299 |
| | TW% | 96 | 95 | 93 | 94 | 97 | 89 | 85 | 95 | 96 |
| Tus | Cars | 4 | 7 | 4 | 4 | 5 | 4 | 5 | 3 | 3 |
| | HS | 15 | 22 | 8 | 15 | 15 | 8 | 9 | 13 | 16 |
| | HS% | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 94 |
| | TW | 480 | 870 | 270 | 600 | 900 | 330 | 270 | 480 | 539 |
| | TW% | 92 | 98 | 89 | 89 | 95 | 89 | 93 | 94 | 76 |

For these instances, we also compute performance measures of our suggested covering plan: how many hot spots we are going to cover and how long are hot spots going to be covered. In Table 3, we present a detailed plan with respect to how many state troopers are needed per shift, per day and per region, shown in row "Cars" and performance measures shown in rows "HS%" and "TW%" for Jefferson, Mobile, and Tuscaloosa areas. From these results, we observe that hot spot coverage percentages are quite close to 100% for our suggested plan. Furthermore, the objective coverage percentage is above 85%, except for three instances. For instance, the "TW%" is 63% for Jefferson ME shift and 76% Tuscaloosa SE shift. This is a factor of start time of the hot spots and travel time required to reach these hot spots. For these instances, even with unlimited resources, it is not possible to fully cover the total hot times, unless the state troopers are allowed to start patrolling from locations other than the state trooper post.

In a final experiment, we evaluate the impact of having hot spots with varying weights. Until this last experiment, all of the experiments assume equally weighted hot spots. However, in real life, some hot spots are more important than others due to the potential severity of the accidents at those locations. We represent these severity levels by attaching different weights to hot spots. We use two arbitrary weight schemes for testing purposes: high variance with weights of 1, 1.5, and 2; and low variance with weights of 1, 1.1, and 1.2. In Table 4, we report the performance of LS with 2, 4, 6, and 8 cars with respect to these two weight schemes. At the bottom row of the table, we calculate the average and maximum gap over all of the instances given a particular resource level. Since TS has similar performance as LS, for the sake of the brevity, we do not report the results. The results of weighted schemes demonstrate the benefit of heuristics, as the heuristics beat the LB of the CPLEX with high percentages, especially for instances with high number of hot spots such as Mobile SA (21 HS), Jefferson MA (27 HS), Jefferson MM (21 HS), and Tuscaloosa MA (22 HS). The benefits are more pronounced with high variance weight scheme. Even though Proposition 1 does not hold for hot spots with varying weights and the heuristics are based on this proposition, the performance of the heuristics is very robust.

Table 4: LS performances for real data with different weight

| | Instances | High Weights (1,1.5,2) | | | | Low Weights (1,1.1,1.2) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 2 | 4 | 6 | 8 | 2 | 4 | 6 | 8 |
| Jeff | MM | 0% | 9% | 24% | 0% | -2% | 0% | 1% | 8% |
| | MA | 22% | -6% | 26% | 30% | 6% | 8% | 13% | 7% |
| | ME | 10% | 0% | 0% | 10% | 0% | 1% | 1% | 2% |
| | FM | -9% | 24% | 24% | 1% | -13% | -4% | -3% | -3% |
| | FA | 0% | -6% | 23% | 11% | -4% | -3% | -1% | -3% |
| | FE | 13% | 0% | 56% | 14% | 3% | 0% | 4% | 4% |
| | SM | -4% | 6% | -19% | -5% | -10% | -5% | -15% | -15% |
| | SA | -7% | -26% | -20% | -15% | -1% | 6% | -12% | -12% |
| | SE | 5% | -1% | 1% | 4% | -2% | 12% | -3% | -3% |
| Mob | MM | 10% | 29% | 1% | 3% | -4% | 2% | 4% | 4% |
| | MA | 6% | 17% | 0% | 21% | -13% | 23% | 2% | 2% |
| | ME | 0% | -8% | 2% | 8% | -8% | -1% | -4% | -4% |
| | FM | -10% | -7% | -33% | -24% | -6% | 5% | -3% | -3% |
| | FA | 2% | 10% | -2% | 4% | -6% | 6% | 0% | 0% |
| | FE | 20% | 2% | 2% | 5% | -5% | -4% | -11% | -11% |
| | SM | 8% | -8% | -5% | -10% | 4% | 13% | 11% | 11% |
| | SA | 17% | 33% | 15% | 15% | -2% | 7% | 2% | 2% |
| | SE | 16% | 0% | 0% | 17% | 1% | 0% | 2% | 2% |
| Tus | MM | -5% | -14% | 15% | -3% | -12% | 10% | -13% | -13% |
| | MA | 25% | 10% | 18% | 8% | 0% | 23% | 6% | 5% |
| | ME | -1% | 31% | 18% | 13% | -9% | 3% | 0% | 0% |
| | FM | -4% | -12% | -12% | -2% | -12% | -9% | -11% | -11% |
| | FA | 30% | 38% | 41% | 35% | 3% | 5% | 7% | 7% |
| | FE | 21% | 39% | 39% | 24% | 0% | 4% | 4% | 4% |
| | SM | -13% | -1% | -1% | -10% | -12% | -11% | -11% | -11% |
| | SA | 3% | 6% | -11% | 4% | -8% | -7% | -9% | -9% |
| | SE | 7% | 7% | 39% | 9% | -11% | 2% | 3% | 3% |
| | Avg. | 6% | 6% | 9% | 6% | -5% | 3% | -1% | -1% |
| | Max. | 30% | 39% | 56% | 35% | 6% | 23% | 13% | 11% |

## 1.6 CONCLUSIONS AND FUTURE WORK

To maximize the effectiveness of state trooper patrols by covering hot spots, we develop a novel model. In this model, we determine whether a state trooper visits a hot spot or not and their arrival and departure times at the hot spots. As the large instances of the problem are beyond the capability of any off-the-shelf optimization software, we design local and tabu search-based algorithms with different neighborhoods. Then, we test our model and solution approaches by using random and real data sets. Compared with the LB of CPLEX, in most instances, our solutions are at least as good as, or better than CPLEX with low runtimes. Furthermore, we have found several instances where CPLEX failed to solve the problem.

The computational testing results are particularly useful for decision-makers in determining the optimal number of state troopers needed for the best coverage. This is important as better coverage is believed to lead to fewer accidents, lower economic impact, and better road safety for everybody. On the other hand, the model also shows the best coverage given a particular resource level. This analysis would be valuable to determine how to reallocate resources in the event of a potential budget cut or increase.

The contributions of the paper to the literature are three-fold. First, the current literature on TOPTW focuses on benefit collection of fixed values given a priori, whereas the MCPRP treats profits as a set of "continuous decision variables" and multiple visits to the same hot spot are allowed. Second, the solution approaches developed can solve even the real-life instances of the problem under a few seconds. Finally, significantly different from the TOPTW literature, this paper introduces *effective patrolling measures* (HS% and TW%) that are useful for decision-makers to determine the optimal levels of coverage for a particular resource level.

Several potential extensions are worthwhile to mention. First, in this paper, we assumed constant travel speed for state troopers traveling from one hot spot to another. Instead of constant travel speed, generalizing the problem where travel speed is correlated with time of day or day of week would be very practical and interesting. Secondly, another extension of the current model is to consider multiple state trooper posts or if state troopers take state trooper cars back home instead of going to the state trooper post. This problem would be analogous to the multi-depot vehicle routing problem with time windows. Thirdly, we are interested in incorporating on-call response into the model, especially to utilize coverage for accidents immediately using dynamic crash information. Finally, the mission statements of many of the highway patrol departments in the United States reflect the belief that issuing citations is an effective auto crash countermeasure (Steil and Parrish, 2009). Hence, the results of this paper can be extended into a revenue management focused application.

## REFERENCES

Anderson, T. 2006. Comparison of spatial methods for measuring road accident "hotspots": a case study of London. *Journal of Maps* 55–63.

Archetti, C., A. Hertz, M. Speranza. 2007. Metaheuristics for the team orienteering problem. *Journal of Heuristics* **13** 49–76.

Birge, J. R., S. M. Pollock. 1989. Modelling rural police patrol. *Journal of the Operational Research Society* 41–54.

Blincoe, L., A. Seay, E. Zaloshnja, T..Miller, E. Romano, S.Luchter, R.Spicer. 2002. The economic impact of motor vehicle crashes, 2000. Crash Statistics DOT HS 809 446, NHTSA.

Boussier, S., D. Feillet, M. Gendreau. 2007. An exact algorithm for team orienteering problems. *4OR: A Quarterly Journal of Operations Research* **5**(3) 211–230.

Braysy, O., M. Gendreau. 2005a. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science* **39**(1) 104–118.

Braysy, O., M. Gendreau. 2005b. Vehicle routing problem with time windows, part II: Metaheuristics. *Transportation Science* **39**(1) 119–139.

Butt, S., T. Cavalier. 1994. A heuristic for the multiple tour maximum collection problem. *Computers and Operations Research* **21** 101–111.

Chaiken, J. M., P. Dormont. 1978. A patrol car allocation model: Capabilities and algorithms. *Management Science* **24**(12) 1291–1300.

Chao, I., B. Golden, E. Wasil. 1996a. A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research* **88** 475–489.

Chao, I., B. L. Golden, E. A. Wasil. 1996b. The team orienteering problem. *European Journal of Operational Research* **88**(3) 464–474.

Chawathe, S. S. 2007. Organizing hot-spot police patrol routes. *Intelligence and Security Informatics, 2007 IEEE* 79–86.

Chen, H. C., M. A. Quddus. 2003. Applying the random effect negative binomial model to examine traffic accident occurrence at signalized intersections. *Accident Analysis and Prevention* **35** 253–259.

Cheng, W., S. P. Washington. 2005. Experimental evaluation of hotspot identification methods. *Accident Analysis and Prevention* **37**(5) 870–881.

Church, R. L., C. ReVelle. 1974. The maximal covering location problem. *Papers of Regional Science Association* **32** 101–118.

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

Duchenne, E., G. Laporte, F. Semet. 2005. Branch-and-cut algorithms for the undirected m-peripatetic salesman problem. *European Journal of Operational Research* **162**(3) 700–712.

Duchenne, E., G. Laporte, F. Semet. 2007. The undirected m-peripatetic salesman problem: polyhedral results and new algorithms. *Operations Research* **55**(5) 949–965.

Feillet, D., P. Dejax, M. Gendreau. 2005. Traveling salesman problem with profits. *Transportation Science* **39** 188–205.

FMSCA. 2008. A guide for planning and managing the evaluation of a tact program. Tech. rep., Department of Transportation, Washington DC.

Gatrell, A. C., T. C. Bailey, P. J. Diggle, B. S. Rowlingson. 1996. Spatial point pattern analysis and its application in geographical epidemiology. *Transactions of the Institute of British Geographers* **21**(1) 256–274.

Golden, B. L., L. Levy, R. Vohra. 1987. The orienteering problem. *Naval Research Logistics* **34**(3) 307–318.

Green, L. 1984. A multiple dispatch queueing model of police patrol operations. *Management Science* **30** 653–664.

Ke, L., C. Archetti, Z. Feng. 2008. Ants can solve the team orienteering problem. *Computers and Industrial Engineering* **54** 648–665.

Laporte, G., S. Martello. 1990. The selective travelling salesman problem. *Discrete Applied Mathematics* **26** 193–207.

Larson, R. 1973. A hypercube queueing model for facility location and redistricting in urban emergency services. Tech. rep., New York City Rand Institute.

Lauri, F., A. Koukam. 2008. A two-step evolutionary and ACO approach for solving the multi-agent patrolling problem. *IEEE Congress on Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence)*. 861–868.

Liang, Y. C., S. Kulturel-Konak, A. E. Smith. 2002. Meta heuristics for the orienteering problem. *Proceedings of the 2002 congress on Evolutionary Computation*, vol. 1. IEEE, 384–389.

Machado, A., G. Ramalho, J. D. Zucker, A. Drogoul. 2003. Multi-agent patrolling: An empirical analysis of alternative architectures. *Lecture Notes in Computer Science* **2581** 81–97.

Marianov, V., C. ReVelle. 1995. Siting emergency services. Z. Drezner, ed., *Facility Location*. Springer, Berlin, 199–223.

McCullagh, M. J. 2006. Detecting hotspots in time and space. *Proceedings of International Symposium and Exhibition on Geoinformation*. 1–18.

Miranda-Moreno, L. F., A. Labbe, L. Fu. 2007. Bayesian multiple testing procedures for hotspot identification. *Accident Analysis and prevention* **39**(6) 1192–1201.

Mitchell, P. S. 1972. Optimal selection of police patrol beats. *The Journal of Criminal Law, Criminology, and Police Science* **63**(4) 577–584.

Montemanni, R., L. Gambardella. 2009. Ant colony system for team orienteering problems with time windows. *Foundations of Computing and Decision Sciences* **34**(4).

National Highway Traffic Safety Administration (NHTSA). 2010. Traffic safety facts. Crash Statistics DOT HS 811 291, NHTSA.

Reis, D., A. Melo, A. L. V. Coelho, V. Furtado. 2006. Towards optimal police patrol routes with genetic algorithms. *Lecture Notes In Computer Science* **3975** 485–491.

Righini, G., M. Salani. 2006. Dynamic programming for the orienteering problem with time windows. Technical Report 91, Universita degli Studi, Milano, Crema, Italy.

Righini, G., M. Salani. 2009. Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming. *Computers and Operations Research* **36**(4) 1191–1203.

Santana, H., G. Ramalho, V. Corruble, B. Ratitch. 2004. Multi-agent patrolling with reinforcement learning. *AAMAS-2004– Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, vol. 3. 1122–1129.

Schilde, M., K. F. Doerner, R. F. Hartl, G. Kiechle. 2009. Metaheuristics for the bi-objective orienteering problem. *Swarm Intelligence* **3**(3) 179–201.

Souffriau, W., P. Vansteenwegen, G. Vanden Berghe, D. Van Oudheusden. 2010. A path relinking approach for the team orienteering problem. *Computers and Operations Research* **To Appear**.

Steil, D., A. Parrish. 2009. Hit: A gis-based hotspot identification taxonomy. *International Journal of Computers and their Applications* **16**(2) 1–10.

Tang, H., E. Miller-Hooks. 2005. A tabu search heuristic for the team orienteering problem. *Computers and Operations Research* **32** 1379–1407.

Tasgetiren, M. F. 2001. A genetic algorithm with an adaptive penalty function for the orienteering problem. *Journal of Economic and Social Research* **4**(2) 1–26.

Tricoire, F., M. Romauch, K. F. Doerner, R. F. Hartl. 2010. Heuristics for the multi-period orienteering problem with multiple time windows. *Computers and Operations Research* **37** 351–367.

Tsiligirides, T. 1984. Heuristic methods applied to orienteering. *Journal of the Operational Research Society* **35**(9) 797–809.

Vansteenwegen, P., W. Souffriau, D. Van Oudheusden. 2010. The orienteering problem: A survey. *European Journal of Operational Research* **To Appear**.

Vansteenwegen, P., W. Souffriau, G. Vanden Berghe, D. Van Oudheusden. 2009. Iterated local search for the team orienteering problem with time windows. *Computers and Operations Research* **36**(12) 3281–3290.

Calvo, R. W., R. Cordone. 2003. A heuristic approach to the overnight security service problem. *Computers and Operations Research* **30**(9) 1269–1287.

## APPENDIX

## PROOF OF THEOREM 1

The maximal covering location problem (MCLP) establishes a set of $m$ facilities to maximize the total weight of "covered" customers, where a customer is considered covered if she is located at most certain specified distance $r$ away from the closest facility. The problem was originally introduced by Church and ReVelle (1974) and is NP-hard (Marianov and ReVelle, 1995). To prove that MCPRP is NP-hard, we need to show that the MCLP is polynomially reducible to MCPRP.

Suppose we had a polynomial algorithm for solving the decision version of the MCPRP. Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, time windows associated with all hot spots, $|\mathcal{K}|$ state trooper cars, shift duration $\mathcal{T}$, and a positive number $\mathcal{B}$, our algorithm would produce a "yes" or "no" answer in polynomial time to the decision question of MCPRP: are there $|\mathcal{K}|$ routes satisfying the time window restrictions of all hot spots and take less than $\mathcal{T}$ such that the total coverage time is at least $\mathcal{B}$? Now, construct an instance of MCPRP as follows: $[e_i,\ l_i] = [e_i,\ e_i + \alpha_i]$, where $\alpha_i$ is an arbitrary small number, say 1 minute, such that the stop at hot spot $i$ can only collect $\alpha_i$.

Now, consider the following notation for the MCLP:

- $\mathcal{I}$ Set of hot spots.
- $\mathcal{J}$ Set of all of the routes that satisfy the time windows and shift duration restrictions.
- $a_i$ Coverage benefit, i.e. for any $k \in \mathcal{K}$, $a_i = f_{ik} - s_{ik} = l_i - e_i = \alpha_i$.
- $N_i$ Set of routes that include hot spot $i$.
- $X_j$ decision variable, 1 if route $j$ is selected as a part of patrolling plan; 0 otherwise.
- $Y_i$ decision variable, 1 if hot spot $i$ is covered; 0 otherwise.

The mathematical formulation is presented as

$$\max \sum_{i \in \mathcal{I}} a_i Y_i \tag{17}$$

$$s.t.$$

$$\sum_{j \in N_i} X_j \geq Y_i, \qquad \forall i \in \mathcal{I}. \tag{18}$$

$$\sum_{j \in \mathcal{J}} X_j = |\mathcal{K}|. \tag{19}$$

$$Y_i \in \{0,1\} \text{ and } X_j \in \{0,1\}, \qquad \forall i \in \mathcal{I} \text{ and } j \in \mathcal{J}. \tag{20}$$

Constraints 18 allow the coverage $Y_i$ to equal 1 only when one or more routes in set $N_i$ are chosen. The number of routes is restricted to $|\mathcal{K}|$ in Constraint 19. The solution to this problem specifies not only the maximal hot spot coverage but also the $|\mathcal{K}|$ routes that achieve this maximal coverage.

The transformation to MCLP is polynomial since all of the problem parameters can be obtained in polynomial time, including the set $\mathcal{J}$. Note that the size and construction of the routes are limited by the time windows of hot spots and the shift duration. If a hot spot is chosen for a route, there are only $(n - p_1)$ choices where $p_1 \geq 1$ due to the time window restrictions, and every time a hot spot is included in a route, the available choices decrease super-linearly. Then, a route can be constructed by evaluating $n \times (n-p_1) \times (n-p_2) \times \ldots \times (n - p_k)$, where $p_k > p_{k-1} > \ldots > p_2 > p_1$ and $p_k < n$ due to $\mathcal{T}$ and time windows. Thus, the set $\mathcal{J}$ can be constructed by an algorithm with $O(n^{p_k})$ complexity.

Overall, the optimal solution to MCPRP provides an answer (yes/no) to the decision version of the MCLP whether there exists $|\mathcal{K}|$ "facilities" (routes) to cover the "customers" (hot spots) to obtain a benefit that is at least $\mathcal{B}$. Therefore, the proof is complete. ■

## PROOF OF PROPOSITION 1

The proof covers two cases. First case considers the situation where pushing the end of service time at one hot spot does not eliminate any visits to the future hot spots. The second case covers the possibility of reduction in the number of hot spots visited in the remainder of the coverage due to incrementing the service time at one hot spot.

**Case 1: No Hot Spot Elimination**

First, let $S^*$ be an optimal solution with the objective function value $v(S^*)$. For route $k \in \mathcal{K}$ in $S^*$, let $i$ be the last hot spot where $f_{ik} < \min(l_i,\ 480 - t_{i,n+1})$.

For a state trooper to get back to the state trooper post at the end of the shift on time, the finish time at the last hot spot of his route should satisfy $f_{ik} + t_{i,n+1} \leq 480$. Now, let us create a new solution $S'$ from $S^*$ where everything is kept the same except $f'_{ik} = \min(l_i,\ 480 - t_{i,n+1})$. Thus, $f'_{ik} > f_{ik}$. Hence, the objective value of $S'$, $v(S')$, is larger than $v(S^*)$, which contradicts that $S^*$ is optimal. Hence, if $i$ is the last hot spot visited on route $k$, $f_{ik} = \min(l_i,\ 480 - t_{i,n+1})$.

Consider now the situation where $i$ is not the last hot spot on route $k$. Suppose $S^*$ is an optimal solution such that there is at least one hot spot $i$ satisfying $f_{ik} < l_i$. We again create a new solution $S'$ from $S^*$ where everything is kept the same except $f_{ik} = l_i$. The difference between $v(S^*) - v(S') = f_{ik} - l_i - s_{i+1,k} + s'_{i+1,k}$, where $s'_{i+1,k}$ is the start time at hot spot $i+1$ on route $k$ in solution $S'$. Now, $s'_{i+1,k} - s_{i+1,k} = \max(l_i + t_{i,i+1},\ e_{i+1}) - \max(f_{ik} + t_{i,i+1},\ e_{i+1})$

$$= \begin{cases} l_i - f_{ik} & \text{if } e_{i+1} \leq f_{ik} + t_{i,i+1}; \\ l_i + t_{i,i+1} - e_{i+1} & \text{if } f_{ik} + t_{i,i+1} < e_{i+1} \leq l_i + t_{i,i+1}; \\ 0 & \text{if } e_{i+1} > l_i + t_{i,i+1}. \end{cases}$$

Note that in all cases, $s'_{i+1,k} - s_{i+1,k} \leq l_i - f_{ik}$. Therefore, $v(S^*) - v(S') < 0$, which contradicts that $S^*$ is the optimal solution. Since $i$ is an arbitrary hot spot, in the optimal solution, $f_{ik} = l_i$ on a route $k$.

**Case 2: Possible Hot Spot Elimination**

In this case, in the newly created solution $S'$, the adjustment at the previous hot spot makes it infeasible to reach the next hot spot(s) on the original route. So, state trooper $k$ needs to skip some hot spot(s) on the original route to go to the next reachable hot spot. We prove this case by induction.

**Case 2a: Base Step** The increment of service time at hot spot $i$ only eliminates the next hot spot $i+1$ on the route. We assume that the triangular inequality holds, that is, $t_{i,i+2} \leq t_{i,i+1} + t_{i+1,i+2}$. Then, for route $k$, the difference in the objective functions $v(S^*)$ and $v(S')$ comes from the changes of contributions of hot spots $i$, $i+1$, and $i+2$. These contributions are

- $\Delta_i = f_{ik} - s_{ik}$ and $\Delta'_i = l_i - s_{ik}$;
- $\Delta_{i+1} = l_{i+1} - \max(e_{i+1}, f_{ik} + t_{i,i+1})$ and $\Delta'_{i+1} = 0$; and
- $\Delta_{i+2} = l_{i+2} - \max(e_{i+2}, l_{i+1} + t_{i+1,i+2})$ and $\Delta'_{i+2} = l_{i+2} - \max(e_{i+2}, l_i + t_{i,i+2})$.

Then, $v(S') - v(S^*) = \sum_{j=i}^{i+2} \Delta'_j - \sum_{j=i}^{i+2} \Delta_j = l_i - \max(e_{i+2}, l_i + t_{i,i+2}) - f_{ik} - l_{i+1} + \max(e_{i+1}, f_{ik} + t_{i,i+1}) + \max(e_{i+2}, l_{i+1} + t_{i+1,i+2})$. Based on different cases of $\max(e_j, l_{i+1} + t_{i+1,j}) - \max(e_j, l_i + t_{ij})$, we simplify this statement and observe that $v(S') \geq v(S^*)$ for every case. Even though one less hot spot is covered, the coverage time is not shortened. Hence, the objective value is at least as good as the original objective value.

**Case 2b: Induction Step** Now, we assume that the increment in the service time eliminates the next consecutive $b > 1$ hot spots. In this case, let $v(S'_b)$ denote the objective function for the modified solution $S'_b$. We assume that $v(S'_b) - v(S^*) \geq 0$. We need to prove that if $b+1$ hot spots are eliminated, $v(S'_{b+1}) \geq v(S_{b+1})$ holds. From the triangular inequality, we know $t_{i,i+b+1} \leq t_{i,i+b+1} + t_{i+b+1,i+b+2} \leq t_{i,i+1} + t_{i+1,i+2} + ... + t_{i+b-1,i+b} + t_{i+b,i+b+1} + t_{i+b+1,i+b+2}$. In addition, for $j = 1, \ldots, b+1$, $\Delta_{i+j} = l_{i+j} - \max(e_{i+j}, f_{ik} + t_{i,i+j})$ and $\Delta'_{i+j} = 0$; and $\Delta_{i+b+2} = l_{i+b+2} - \max(e_{i+b+2}, l_{i+b+1} + t_{i+b+1,i+b+2})$ and $\Delta'_{i+2} = l_{i+b+2} - \max(e_{i+b+2}, l_i + t_{i,i+b+2})$. Then,

$$
\begin{aligned}
v(S'_{b+1}) - v(S^*) &= \sum_{j=i}^{i+b+2} \Delta'_j - \sum_{j=i}^{i+b+2} \Delta_j \\
&= v(S'_b) - v(S^*) - (l_{i+b+1} - \max(e_{i+b+1}, l_i + t_{i,i+b+1})) \\
&\quad - \max(e_{i+b+2}, l_i + t_{i,i+b+2}) + \max(e_{i+b+2}, l_{i+b+1} + t_{i+b+1,i+b+2})
\end{aligned}
$$

Based on the cases of $\max(e_{j+1}, l_j + t_{j,j+1}) - \max(e_{j+1}, l_i + t_{i,j+1})$ and the induction step, $v(S'_{b+1}) \geq v(S^*)$. Hence, the modified solution is as good as $S^*$.

This concludes the proof. ■

# 2: BI-CRITERIA DYNAMIC LOCATION-ROUTING PROBLEM FOR PATROL COVERAGE

## 2.1 INTRODUCTION

Accidents on the road are tragedies to individuals who are involved as well as the society as a whole. Light accidents can lead to traffic jams which bring inconvenience to citizens traveling on the road or damage personal possessions and public infrastructure; while heavy accidents are even more traumatic, taking lives away or leaving people with life-long impairment such as paralysis. No matter how light or serious accidents are, we do not want to see any of them happen, and the ultimate goal is to achieve zero accidents.

Due to both economic and humanitarian importance, maintaining roadway travel safety has aroused widespread interest from all levels of the society, including citizens, government officials, industry practitioners, and academicians. The National Highway Traffic Safety Administration estimates the cost of improving safety by way of various law enforcements as high as $230.6 billion, a year-nearly 2.3 percent of the nation's gross domestic product (Blincoe et al., 2002). At the same time, academicians (Steil and Parrish, 2009; Keskin et al., 2011; Lou et al., 2011; Willemse and Joubert, 2012) are also rigorously pushing forward more effective law enforcement plans, including effective patrolling plans. One way of improving patrolling efficiency is focusing on patrolling critical locations with high crash frequencies.

Given historical crash data, a "hot spot"(HS) is defined as a certain stretch of highways with high frequency of crashes of different severity levels over a certain time period. In this problem, we are interested in finding the right start and stop locations (temporary stations) for state troopers at the beginning and end of their shift, respectively, as well as the patrol routes to visit time-critical HSs. Our overall goal is to maximize the visibility of the state troopers during the *hot times* of the HSs while minimizing the costs associated with utilization of troopers, traveling from one HS to another, and potential fees for temporary stations. Therefore, we tackle a bi-criteria optimization problem.

Specifically, we assume that at the beginning of the shift, state troopers start their patrol at temporary stations whose locations need to be determined from a list of potential stations. During their shift, starting at their selected temporary station, the state troopers travel from one HS to another and stop at a HS for a while during the effective coverage time, i.e. during the time interval that particular location is critical. At the end of the shift, the state troopers go to other temporary stations so that the travel time from the last covered HS in the previous shift and the travel time to the next HS in the next shift is optimized. The locations of the ending temporary stations need to be determined as well. With these characteristics, this problem is similar to a *multi-depot (multiple temporary stations), dynamic location (changing locations) and routing problem* (LRP). At the same time, since we are aiming to maximize the presence of the state troopers at the defined HSs and the service time at a hot spot can be viewed as the "variable profit," the problem has resemblance to the *team orienteering problem with time windows (TOPTW).*

We note that our paper is closely related to the work by Keskin et al. (2011) that focuses on patrol coverage of hot spots. Considering a single station, Keskin et al. (2011) proposes a mixed integer linear optimization model, called the maximum covering patrol routing problem (MCPRP), to maximize the presence of state troopers at the defined HSs for a given patrol shift. They show that the problem of interest is related to the TOPTW and prove that the MCPRP is $NP$-hard. They develop efficient local search- and tabu search-based heuristics to solve real life instances. In their results, they note that despite the effectiveness of the solutions and regardless of the number of state troopers, it is not possible to cover all of the time-sensitive HSs by just starting from a single station. HSs are geographically dispersed and time sensitive. By the time the state troopers reach a distant HS location, the effective coverage would have already lapsed. Our work extends their paper in three directions:

(i) We consider multiple temporary stations whose locations need to be determined as opposed to a single depot. This way, we can cover more HS which are located out of the accessibility range with just one station.

(ii) Our model spans multiple periods (shifts) as the locations of the HSs and temporary stations dynamically change and temporary station locations tie the multiple periods together.

(iii) In addition to "coverage benefit" maximization, we also consider the minimization of total system costs (cost of utilization of troopers, travel costs, and temporary station location costs). With the addition of temporary stations, the coverage is expected to go up. But, it is also important to account for how much this coverage is going to cost. The costs included in the analysis create an immediate trade-off with respect to resource utilization and hot spot coverage. For instance, if less state troopers are dispatched or less temporary stations are located, the state troopers need to travel farther and spend more time on the road rather than covering HSs. On the other hand, if more state troopers are dispatched and more temporary stations are opened, there may not be enough monetary resources to pay for patrolling costs.

Since the MCPRP, which arises as a subproblem for our problem, and dynamic location-routing problem are shown to be NP-hard, we resort to heuristic approaches. We first present a mixed integer programming formulation of the problem that can be solved via off-the-shelf software. Then, we develop efficient, tailored heuristics based on effective neighborhood searches embedded within a simulated annealing framework which simultaneously allows opening/closing temporary stations (TS), increasing/decreasing state troopers, and adding/dropping HS. When we compare the tailored heuristics with the off-the-shelf software, we see that our solutions provide good quality solutions in short periods of time. Additionally, we provide additional service measures including the percentage of number of hot spots covered and percentage of coverage length based on the outcome of the heuristics. These service measures provide additional insights into the solutions and help in evaluating the constraints related to the number of state trooper cars and patrol duration.

The remainder of this paper is structured as follows: in Section 2.2, we present the literature review. In Section 2.3, the details of the general mathematical model are discussed, including necessary assumptions and notations. Next, in Section 2.4, we present the analysis of the problem and the solution approaches based on the characteristics of the problem. In Section 2.5, we discuss the computational results based on the heuristics and exact approaches. Finally, in Section 2.6, we conclude with the summary and recommendations.

## 2.2 LITERATURE REVIEW

As our problem has similarities to TOPTW and LRP, we review both of these areas.

### 2.2.1 TOPTW

The OP is first introduced by Tsiligirides (1984) for the orienteering competition. The goal is to find a circuit that maximizes collected profit such that travel costs do not exceed a preset value $C$. Some of its important variants include the team orienteering problem (TOP)–where a fixed number of paths is considered, the orienteering problem with time windows (OPTW), and the team orienteering problem with time windows (TOPTW).

Boussier et al. (2007), Montemanni and Gambardella (2009), and Vansteenwegen et al. (2009) are the only known people to solve TOPTW. Boussier et al. (2007) propose an exact branch-and-price algorithm, which is generic enough to handle different kinds of OP, including the TOPTW. Montemanni and Gambardella (2009) develope local search and ant colony system algorithms based on the solution of a hierarchic generalization of TOPTW. Lastly, Vansteenwegen et al. (2009) present a straightforward and very fast iterated local search heuristic, which combines an insertion step and a shaking step– reverse insertion operation, to escape from local optima. Note that all of these papers only consider single period problems. To our best knowledge, only Tricoire et al. (2010) work on a multi-period OPTW problem, and solve it through a variable neighborhood search based metaheuristic.

However, they solve only one car instance and specify a fixed starting depot and a fixed stopping depot for each period; whereas in our case, we solve for multiple state troopers and the starting and ending locations are decision variables.

### 2.2.2 LRP

LRP receives little attention until Salhi and Rand (1989) show that LRP consistently produces better solution than solving sub-problems of facility location and vehicle routing sequentially. In the first LRP survey paper, Laporte (1988) summarizes two-index or three-index vehicle flow formulations for static, deterministic LRP. Later on, more reviews by Balakrishnan et al. (1987), Min et al. (1998), and Nagy and Salhi (2007) bring developments in LRP up to date.

Both exact algorithms and heuristics are designed to solve LRP problem, but exact algorithms (see Labbé et al. (2004) and Laporte et al. (1986)) are still limited to small to medium size problems and heuristics are far more prevalent. Nagy and Salhi (2007) categorize heuristics into sequential, clustering, iterative (Hansen et al. (1994), Perl and Daskin (1985), and Wu et al. (2002)), and hierarchical heuristics (Albareda-Sambola et al. (2007), Melechovskỳ et al. (2005), and Nagy and Salhi (1996)). As sequential and clustering heuristics fail to utilize feedback information between location part and routing part, iterative and hierarchical heuristics are more commonly seen done in the literature. Especially when a problem has obvious hierarchy involved, hierarchical heuristic works the best. It divides LRP into a master problem of facility location, and its subordinate problem of vehicle routing. We follow this pattern in the development of our heuristics.

We note two important differences between our work and the literature. Firstly, our problem has time window limitation, to the best of our knowledge, which has not been addressed in LRP literature before. Even though Nagy and Salhi (2007) point out in their survey that work by Semet and Taillard (1993) belongs to this category, that paper should be viewed as VRPTW literature instead of LRP since there were no location decisions. Secondly, instead of locating long-term depots, we locate temporary stations while optimizing routing schedules. For our problem, both location and routing are short-term decisions, avoiding elegantly the common criticism that LRP has conflicting planning horizons in the short and the long run.

## 2.3 GENERAL MODEL

### 2.3.1 Problem Definition

Given the problem specifications earlier, it is assumed that at the beginning of each shift, state trooper cars are dispatched from temporary stations (TSs), where the potential locations are given as $i \in \mathcal{I} = \{1, 2, ..., |\mathcal{I}|\}$. In addition to TSs, state trooper cars can be also dispatched from state trooper posts, so the set of state trooper posts $\mathcal{A}$ is a subset of $\mathcal{I}$. $\mathcal{K} = \{1, 2, ..., |\mathcal{K}|\}$ is the set of the available state troopers, and each trooper on duty incurs a cost of $v$ (\$/shift/trooper). Let $\mathcal{P} = \{1, 2, 3\}$ be the set of shifts, representing morning, afternoon, and night shifts, and $\mathcal{D} = \{1, 2, ..., |\mathcal{D}|\}$ be the set of days, where 1 represents the first day and so on. In fact, pairs of $p \in \mathcal{P}$ and $d \in \mathcal{D}$ can be represented by a single period index $t \in \mathcal{T} = \{0, 1, ..., |\mathcal{P}| \times |\mathcal{D}|\}$.

Within a subset of regions with given potential locations for TS $i \in \mathcal{I}$ and during a particular period $t \in \mathcal{T}$, there are historically established HS, $j = 1, \ldots, \mathcal{N}_t$. In a period $t \in \mathcal{T}$, HS $j$ have three attributes: (i) location on the mile-posted road network; (i) the time window $[e_j^t, l_j^t]$ when HS $j$ becomes "hot" where $e_j^t$ and $l_j^t$ are the start and end times of the "hotness" window; and (iii) weight $w_j^t$ representing severity level. By definition, $e_j^t < l_j^t$. Furthermore, we assume that without loss of generality, $\mathcal{N}_t$ is an ordered set according to $e_j^t$ such that $e_1^t \leq e_2^t \leq \ldots \leq e_n^t$. We define $\mathcal{N} = \cup_{t \in \mathcal{T}} \mathcal{N}_t$. Note that a location can be listed as two different HS as $i$ and $j$, where $e_i^t < e_j^t$ if it becomes "hot" twice within the same period.

Let $\mathcal{V} = \mathcal{N} \cup \mathcal{I}$ denote the union of the sets of HS and locations of potential TS. Additionally, we let $\mathcal{E} = \{(i, j) : \; i, j \in \mathcal{N} \cup \mathcal{I}, i \neq j\}$ define the set of edges. The connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents the underlying road network. $d_{ij}^t > 0$ denotes the shortest travel time from HS $i$ to $j$, $\forall (i,\ j) \in \mathcal{E}$, and in period $t \in \mathcal{T}$. Meanwhile, we define:

- $\triangle^+(i) = \{j \in \mathcal{V},\ t \in \mathcal{T} : (i, j) \in \mathcal{E}, e_i^t + d_{ij}^t \leq l_j^t\}$ as the set of vertices that are directly reachable from $i \in \mathcal{V}$ within the time window, and

- $\triangle^-(i) = \{j \in \mathcal{V},\ t \in \mathcal{T} : (j,i) \in \mathcal{E}, e_j^t + d_{ij}^t \leq l_i^t\}$ as the set of vertices from which $i$ is directly reachable.

The additional assumptions of the model include:

1. The fixed cost of TS is negligible.

2. There is no capacity limit at TS; i.e., multiple state trooper cars can start/stop at the same TS, if desired.

3. Visits of state troopers at HSs are only effective within the time windows of HSs.

4. At the beginning of the shift, a state trooper leaves a selected TS, and at the end of the shift, he may or may not come back to the same TS.

5. State trooper cars travel at a constant speed of 60 miles/hour, thus 1 minute corresponds to 1 mile. This way, distance and time can be easily translated to one other.

6. State troopers can choose whether to visit a HS or not, as well as time to begin and end the coverage. If a HS is chosen by a state trooper, it cannot be visited by others.

Given these setup and notation, we optimize the dynamic selection of TS utilized each period, allocation of state troopers to TS, and routing plan of state troopers simultaneously. Figure 7 shows an example with 5 potential TSs including 2 depots, 3 available state troopers available, 2 periods, and 16 HSs per period. At the beginning, these 3 cars are parked at TS 2, 3, and 5, respectively. The routes, represented by the directed arrows, form a feasible solution while meeting the time windows of the visited HSs. In the first period when $t = 1$, all 3 troopers are utilized; when $t = 2$, only 2 troopers are utilized due to budget limitations. That is, the cost minimization overcomes the benefit maximization. When $t = 1$, $k_1$ starts at TS= 2 but ends at TS= 1, $k_3$ starts at TS= 5 but ends at TS= 3, and only $k_2$ starts and ends at the same TS.

2.3.1.1 Decision Variables

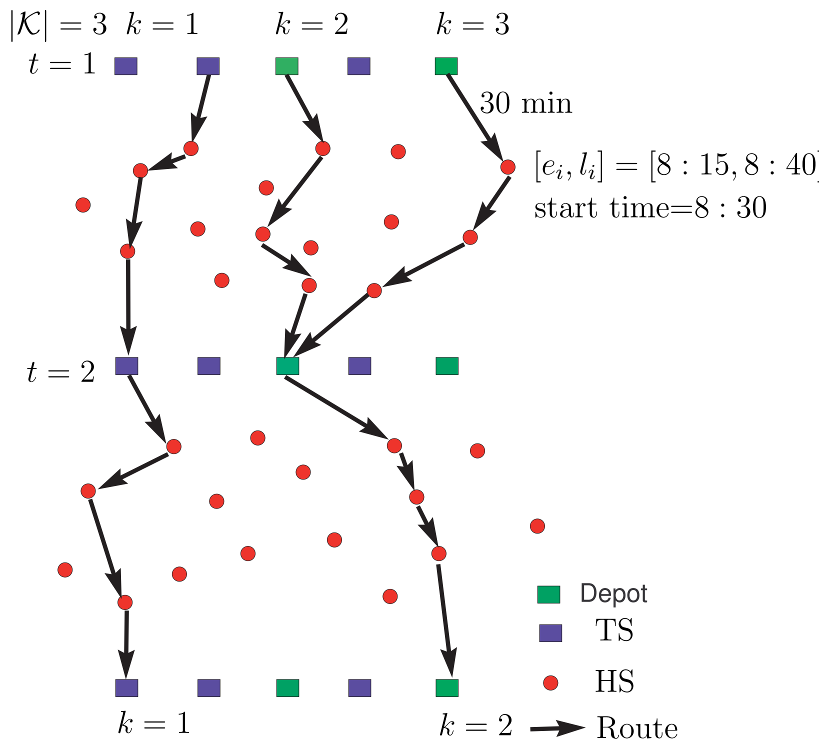



Figure 7: A representative example

We define five sets of decision variables: (i) $x^t_{ijk} = 1$, if state trooper car $k \in \mathcal{K}$ travels from $i$ to $j$, $(i, j) \in \mathcal{E}$ during $t \in \mathcal{T}$, and 0, otherwise. (ii) $s^t_{ik} \geq 0$, the starting time of service for state trooper car $k \in \mathcal{K}$ at HS $i \in \mathcal{V}$ during $t \in \mathcal{T}$. (iii) $f^t_{ik} \geq 0$, the time state trooper car $k \in \mathcal{K}$ leaves HS $i \in \mathcal{V}$ during $t \in \mathcal{T}$, i.e., the end of service. (iv) $y^t_{ik} = 1$ if state trooper $k$ serves $i \in \mathcal{V}$ during $t \in \{\mathcal{T} \cup |\mathcal{T}| + 1\}$, 0, otherwise. (v) $R^t_{ijk} = 1$ if state trooper car $k \in \mathcal{K}$ is relocated from one TS $i$ to another TS $j$ at the end of $t \in \mathcal{T}$, $i, j \in \mathcal{I}$.

2.3.1.2 Objective

We have a multi-objective optimization problem, including cost (trooper utilization cost, routing cost, and facility cost) minimization and benefit (coverage) maximization. All cost parameters are scaled down to the same time span, that is, one shift. Considering all of the factors of vehicle utilization cost, we set $v$ = \$36.63/shift/trooper based on the assumed average wage of \$40,000/year/trooper. On average, fuel price is \$3.5/gallon, and fuel consumption is 0.04 gallon/mile, we set the trip cost as $c$ = \$0.14/mile or \$0.14/minute.

Then, our objective is:

$$\min_{\mathbf{x}} \quad \left( v \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{N}_t} \sum_{k \in \mathcal{K}} x^t_{ijk} + c \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}} d^t_{ij} x^t_{ijk} \right) \tag{21}$$

$$\max_{\mathbf{f},\mathbf{s}} \quad \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{N}_t} \sum_{k \in \mathcal{K}} (f^t_{jk} - s^t_{jk}) \times w_j \tag{22}$$

For multi-objective optimization problems, it is very common that objectives may not be commensurable with each other. Similarly, for our problem, the coverage benefit is measured in minutes whereas the total cost is measured in dollars. Facing this dilemma, the vast majority of researchers used either weighted sum of the objectives or the bounded objective function method or $\varepsilon$-constraint approach. The first group of researchers (Alçada-Almeida et al., 2009; Alumur and Kara, 2007; Caballero et al., 2007) transformed conflicting objectives into a weighted sum of them by attaching each objective with some coefficients as priori. However, due to the arbitrary choices of coefficients, we adopt the other commonly used method: the bounded objective function method or also known as $\varepsilon$-constraint approach (Bérubé et al., 2009; Chankong and Haimes, 1983; Laumanns et al., 2005, 2006; Mavrotas, 2009; Miettinen, 1999). This approach considers the single most important objective and puts all of the other objective(s) into the formulation as constraint(s). Thereafter, our problem is transformed into a benefit maximization problem by setting an upper limit on the budget, say $B$:

$$v \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{N}_t} \sum_{k \in \mathcal{K}} x^t_{ijk} + c \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}} d^t_{ij} x^t_{ijk} \leq B$$

In our computational experiments, we test different levels of $B$ to demonstrate the effect of costs on the patrol routes.

#### 2.3.1.3 Constraints

We categorize our constraints under five groups: schedule feasibility constraints (*1a*)-(*1d*), route structuring constraints (*2a*)- (*2e*), TS updating constraints (*3a*)-(*3d*), car related constraints (*4*), and last but not the least, integrality and non-negativity constraints (*5a*)-(*5b*). In constraints (*1a*), $M_{ij}^t = \max\{l_i^t + d_{ij}^t - e_j^t, 0\} \geq 0$, and in constraints (*3d*), $D_{\text{limit}}$ is a constant that is set to 20 minutes.

Schedule feasibility

$$f_{ik}^t + d_{ij}^t - s_{jk}^t \le (1 - x_{ijk}^t)M_{ij}^t, \quad \forall t \in \mathcal{T},\ (i,j) \in \mathcal{E},\ k \in \mathcal{K}. \tag{1a}$$

$$e_i^t \times \sum_{j \in \triangle^+(i)} x_{ijk}^t \le s_{ik}^t, \quad \forall t \in \mathcal{T},\ i \in \mathcal{V},\ k \in \mathcal{K}. \tag{1b}$$

$$l_i^t \times \sum_{j \in \triangle^+(i)} x_{ijk}^t \ge f_{ik}^t, \quad \forall t \in \mathcal{T},\ i \in \mathcal{V},\ k \in \mathcal{K}. \tag{1c}$$

$$s_{ik}^t \le f_{ik}^t, \quad \forall t \in \mathcal{T},\ i \in \mathcal{V},\ k \in \mathcal{K}. \tag{1d}$$

Route structuring

$$\sum_{i \in \triangle^-(j)} x_{ijk}^t = \sum_{i \in \triangle^+(j)} x_{jik}^t, \quad \forall t \in \mathcal{T},\ j \in \mathcal{N}_t,\ k \in \mathcal{K}. \tag{2a}$$

$$\sum_{j \in \triangle^+(i)} x_{ijk}^t \le y_{ik}^t, \quad \forall t \in \mathcal{T},\ i \in \mathcal{I},\ k \in \mathcal{K}. \tag{2b}$$

$$\sum_{j \in \triangle^-(i)} x_{jik}^t \le y_{ik}^{t+1}, \quad \forall t \in \mathcal{T},\ i \in \mathcal{I},\ k \in \mathcal{K}. \tag{2c}$$

$$\sum_{j \in \triangle^+(i)} x_{ijk}^t = y_{ik}^t, \quad \forall t \in \mathcal{T},\ i \in \mathcal{N}_t,\ k \in \mathcal{K}. \tag{2d}$$

$$\sum_{k \in \mathcal{K}} y_{jk}^t \le 1, \quad \forall t \in \mathcal{T},\ j \in \mathcal{N}_t. \tag{2e}$$

TS updating

$$R_{ijk}^t \le y_{ik}^t, \quad \forall t \in \mathcal{T},\ i,j \in \mathcal{I},\ j \ne i,\ k \in \mathcal{K}. \tag{3a}$$

$$R_{ijk}^t \le y_{jk}^{t+1}, \quad \forall t \in \mathcal{T},\ i,j \in \mathcal{I},\ j \ne i,\ k \in \mathcal{K}. \tag{3b}$$

$$R_{ijk}^t \ge y_{ik}^t + y_{jk}^{t+1} - 1, \quad \forall t \in \mathcal{T},\ i,j \in \mathcal{I},\ j \ne i,\ k \in \mathcal{K}. \tag{3c}$$

$$d_{ij}^t R_{ijk}^t \le D_{\text{limit}}, \quad \forall t \in \mathcal{T},\ i,j \in \mathcal{I},\ j \ne i,\ k \in \mathcal{K}. \tag{3d}$$

Car related

$$\sum_{i \in \mathcal{I}} y_{ik}^t \le 1, \quad \forall t \in \mathcal{T},\ k \in \mathcal{K}. \tag{4}$$

Integrity and non-negativity

$$s_{ik}^t, f_{ik}^t \ge 0, \quad \forall t \in \mathcal{T},\ i \in \mathcal{V},\ k \in \mathcal{K}. \tag{5a}$$

$$x_{ijk}^t, y_{ik}^t, u_{jkg}^t, R_{ijk}^t \in \{0,1\}, \quad \forall t \in \mathcal{T},\ i,j \in \mathcal{V},\ k,\ g \in \mathcal{K},\ g > k. \tag{5b}$$

This model generalizes the MCPRP by Keskin et al. (2011) from single-depot to multi-depot and from a static depot location to dynamic depot locations. In Constraints (*2b*)-(*2c*), $y_{jk}^t = 1$ if state trooper $k$ starts his shift at $j$ in period $t$, and $y_{jk}^{t+1} = 1$ if state trooper $k$ ends his shift at $j$ in period $t$, which is also the starting place for period $t+1$. In the third set of Constraints (*3a*)-(*3c*), $R_{ijk}^t$ can only be equal to 1 when both $y_{ik} = 1$ and $y_{jk}^{t+1} = 1$, or else, $R_{ijk}^t = 0$ for $i \in \mathcal{I}$. This is when state trooper $k$ starts at a TS $i$ and ends at another TS $j \neq i$, where he is relocated from one TS to another. If relocation occurs, the distance between the starting and the stopping TS should not exceed $D_{\text{limit}}$, which is achieved by Constraints (*3d*). This is a practical constraint required by the state troopers. In the fourth set, Constraints (*4*) stipulate that one car can only be parked at one TS.

#### 1.0.1 Overall Model

The overall model is subject to constraints (*1a*)–(*5b*). We call this model, the dynamic multi-depot MCPRPR, in short, DMD-MCPRP.

**Remark 1** *If a state trooper must go back to where he starts his shift, $R_{ijk}^t$ and its related constraints are not needed any more. This is a special case of DMD-MCPRP, which can be solved by period and independently.*

### 2.3.2 Extension with Fixed Charge Considerations

We now consider the case when there is a fixed cost associated with each utilized TS. This case is more applicable if, for instance, each TS is charged with some parking fee, denoted as $F_i$.

To incorporate the fixed costs into the model, another set of decision variables is needed. We define $z_i^t = 1$ if TS $i \in \mathcal{I}$ is open in $t \in \mathcal{T}$, 0, otherwise. With this new variable, the model requires the following two updates. First, the budget constraint has one additional term of the total fixed cost, that is,

$$v\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{N}_t}\sum_{k\in\mathcal{K}}x^t_{ijk} + c\sum_{t\in\mathcal{T}}\sum_{(i,j)\in\mathcal{E}}\sum_{k\in\mathcal{K}}d^t_{ij}x^t_{ijk} + \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}}F_i z^t_i \leq \mathcal{B}.$$

Second, the model is augmented with one additional set of constraints, guaranteeing that TS is marked as open if selected.

$$\text{TS selection} \qquad z^t_i \geq y^t_{ik}, \qquad \forall t\in\mathcal{T},\ i\in\mathcal{I},\ k\in\mathcal{K}. \tag{7}$$

## 2.4 SOLUTION APPROACHES

We observe that both DMD-MCPRP and its extended model are mixed integer linear programs (MILP) and can be solved by CPLEX 12.1[2]. Unfortunately, even for very small instances such as the example in Figure 7, CPLEX runs out of memory.

Among different solution options, we choose a hierarchical heuristic as our problem has an obvious hierarchical structure. As our objective is transformed into a benefit maximization by stating the incurred costs under a budget limit, we first solve the multi-depot MCPRP (MD-MCPRP), then the locations of the temporary stations are determined. This decomposition makes the multi-depot MCPRP problem solvable by period and by shift. Therefore, it provides an opportunity to utilize the solution of Keskin et al. (2011) with slight modifications due to multi-depot considerations. We solve the location problem via a greedy heuristic. We iterate among these two problems to search for better feasible solutions for the overall problem. We first discuss the details of the heuristic for the base model (without TS location costs) and then present a modified heuristic to handle the extended model (with TS fixed costs) next.

### 2.4.1 Heuristic for DMD-MCPRP

[2]CPLEX is a trademark of IBM.

For the base model when the fixed costs of TS are negligible, the optimal solution has the following characteristic:

**Observation 1** *If the optimal hot spot routes in MD-MCPRP are known, the nearest TSs to the first and last HSs in the routes are selected as start and stop locations.*

The proof is provided in the appendix. Based on this observation, we build our heuristic approach. First, note that the first problem is the multi-depot MCPRP that determines the multi-car routing among HSs to maximize the benefits of visiting hot spots. This problem ignores the selection of locations for TSs (depots) and the budget limit temporarily. However, in order to initiate the building of the routes, we need initial starting locations for the routes. For this purpose, we use three initialization strategies: (i) **STR1**: start at the HS with the earliest time window; (ii) **STR2**: start at the HS with the highest weight; (iii) **STR3**: use a combination of STR1 and STR2: that is, out of the first 5 earliest HSs, pick the HS with the highest weight. The heuristic is run using one of these strategies and we report the computational results with different strategies in Section 1.0.1.

The heuristic has six components that include **initialization**, **MCPRP algorithm** (Keskin et al., 2011), **add/drop**, **Select TSs**, **insert/erase**, and **simulated annealing**. The pseudo-code of the algorithm that explains how these components are utilized is given in Display 2. Next, we explain the details of each component.

**Algorithm 2** *$DMD - MCPRP$ heuristic*$(Obj^*,\ Res^*,\ Rou^*,\ Car_t^*)$

**Initialization:** $Obj^* = \infty$, $Res^* = 0$, $Rou^* = \emptyset$, $Car_t^* = 0$. Equally allocate $Car_t^* = \min\{\lfloor \frac{(1-a)\mathcal{B}}{|\mathcal{T}|v} \rfloor, |\mathcal{K}|\}$ cars $\forall t \in \mathcal{T}$. Use a starting strategy to pick starting location for $Car_t^*$, $\forall t \in \mathcal{T}$.
**MCPRP**$[Obj^*,\ Res^*,\ Rou^*]$.
**Add/Drop**$[Obj^*,\ Res^*,\ Rou^*,\ Car_t^*]$.
**Select TSs** $[Obj^*,\ Res^*,\ Rou^*]$.
**if** Budget allows **then**
  **Insert**$[Obj^*,\ Res^*,\ Rou^*]$ between the starting TS and starting HS;
**else**
  **Erase**$[Obj^*,\ Res^*,\ Rou^*,\ Car_t^*]$.
**end if**
**Simulated annealing**$[Obj^*,\ Res^*,\ Rou^*,\ Car_t^*]$.
Return $Obj^*,\ Res^*,\ Rou^*,\ Car_t^*$.

#### 2.4.1.1 Initialization

In the **Initialization** step, we first initialize the objective coverage $Obj^*$, resource consumption level $Res^*$, and the set of route sequence information $Rou^*$. To determine the number of cars available in each period $Car_t^*$, we compare the available budget for employees with the total number of cars. To allocate the budget per car appropriately, we assume that employee salary portion of the budget is divided equally in each period, that is the maximum number of cars in a given period cannot exceed $\lfloor \frac{(1-a)\mathcal{B}}{|\mathcal{T}|v} \rfloor$, where $(1-a)$ is the portion of the budget spent on employee salary. Then, the initial number of available cars in period $t$ is calculated as $\min\{\lfloor \frac{(1-a)\mathcal{B}}{|\mathcal{T}|v} \rfloor, |\mathcal{K}|\}$. Afterwards, using one of the aforementioned starting strategies, we initialize the starting locations of each car.

Given the number of cars and their starting locations, we utilize the MCPRP algorithm developed by Keskin et al. (2011). This algorithm builds $Car_t^*$ many routes in a greedy fashion, improved with exchange and relocate operators.

#### 2.4.1.2 Add/Drop Component

After the MCPRP algorithm, the initial routes are built up for all of the periods. Using this information, we calculate the resource consumption $Res^*$ by taking into account the travel costs incurred by the formed routes. If all the costs have the feature of proportionality, the ratio of gas cost ($x$) over state trooper salary should be equal to the ratio of gas budget percentage and state trooper salary budget percentage, that is, $\frac{x}{v} = \frac{a\mathcal{B}}{(1-a)\mathcal{B}}$. So, $(1 + \frac{a}{1-a})v$ is an approximate cost for utilizing one car ($v$) and traveling to hot spots ($\frac{av}{1-a}$). If the consumed resource level $Res^*$ is less than $\mathcal{B} - (1 + \frac{a}{1-a})v$, and there is an available (unused) state trooper car, we can add one more patrol route to the period with the largest number of uncovered HSs, i.e., $Car_t \leftarrow Car_t + 1$. Until all of the budget is used or all of the state trooper cars are utilized, we keep adding a new patrol route. Each new route is again built using the MCPRP algorithm.

On the other hand, if the total resource consumption $Res^*$ after the initial route construction exceeds the available budget by more than $v$, i.e., $Res^* > \mathcal{B} + v$, we eliminate the routes with the least coverage time until the total budget is controlled.

#### 2.4.1.3 Selecting TS Locations

The next step in the overall algorithm involves selecting TS locations. By observation 1, for each state trooper, we simply pick the closest TSs to the starting and stopping HSs in his route to begin and end his shift as long as the start and end TSs are within distance $D_{\text{limit}}$. In other words, as opposed to considering all of the candidate TS locations, we only consider the ones within distance $D_{\text{limit}}$. We repeat this process for all $t \in \mathcal{T}$. In essence, this step achieves the goal of picking a common TS which has the smallest travel distance from the stopping HS of one period to the starting HS of next period in a myopic fashion. After all of the TS locations are selected, the routes are formed. The pseudo-code is given in Display 3.

**Algorithm 3** Selecting TSs ($Obj^*,\ Res^*,\ Rou^*$)

**for** $t = 0;\ t < |\mathcal{T}|;\ t++$ **do**
    **if** $t = 0$ **then**
        Pick the closest TS to the starting HSs on routes in periods $t$.
    **else**
        **if** $t > 0$ and $t < |\mathcal{T}|$ **then**
            Pick the closest TS to the stopping HSs on routes in periods $t$ and starting HSs on routes in periods $t+1$, which is within $D_{\text{limit}}$ to the chosen TS in period $t-1$.
        **else**
            Pick the closest TS to the stopping HSs on routes in periods $t$.
        **end if**
    **end if**
    Update $Obj^*,\ Res^*,\ Rou^*$.
**end for**

After this component, the heuristic completes a location-routing cycle. However, the budget may still be violated. Therefore, the next two components (**Insert** and **Erase**) improve this location-routing solution by taking the budget limit into account. They are similar to the insertion and shaking steps by Vansteenwegen et al. (2009).

#### 2.4.1.4 Insert/Erase Component

As the traveling from the selected TS locations to the HSs increases the resource consumption, the new resource consumption may exceed the budget limit. If the budget is exceeded, **Erase** keeps deleting the HSs with the least coverage time until the resource consumption is within budget limits. One possible result of this operation is that all of the HSs of a route are removed. If this is the case, then that route does not cover any HS other than TS, and this route is closed. Then, we free up the state trooper car, decrease the number of cars used in that period, and reduce the utilization cost $v$ from the resource consumption. Since now additional resources are available, **Insert** component is called to insert any uncovered HS while considering the travel costs as well as the coverage benefit obtained from the inclusion of this HS.

On the other hand, if the inclusion of travel costs from and to selected TS locations into the resource consumption does not exceed the budget limitation, we may re-call **Insert** component to include uncovered HSs until all of the budget is utilized.

#### 2.4.1.5 Simulated Annealing Component

To optimize the patrol routes and TS locations, we develop a simulated annealing algorithm. Simulated annealing (SA), first proposed by Kirkpatrick et al. (1983), is one of the most well-developed and widely used iterative techniques for solving optimization problems (Sait and Youssef, 1999). The basic requirements of the SA algorithm are a neighborhood structure on the set of feasible solutions and a number of parameters which govern the acceptance or rejection of new solutions generated during the search. In our SA implementation, we utilize relocate and exchange neighborhoods to improve the routes by considering different HS inclusions.

SA is a randomized search method that tries to improve a solution by a random walk in the solution space and gradually adjusting a parameter called *temperature*. The sequence of temperatures and the number of iterations for which they are maintained is called the *annealing schedule*. The quality of the solution is very sensitive to both of these factors. Therefore, the SA algorithm requires an initial temperature, $T_0$; a cooling rate, $\alpha$; a progressive factor, $\beta$; the total allowed time for the annealing process, $MaxTime$; and, finally, the time until the next parameter update, $M$ (Sait and Youssef, 1999, pages 53-55). In our implementation, we experimented extensively to find an effective combination of these parameters. We set $T_0 = 1000$, $\alpha = 0.9$, $\beta = 2$, $MaxTime = 8000$ seconds, and $M = 2$. The details of the simulated annealing metaheuristic are given in Display 4.

**Algorithm 4** Procedure $SA(Obj^0, Res^0, Rou^0, Car_t^0, T_0, \alpha, \beta, M, MaxTime)$

$T = T_0$.
$Res^{current} = Res^0$; $Rou^{current} = Rou^0$; $Car_t^{current} = Car_t^0$; $Obj^{current} = Obj^0$.
$Time = 0$.
**while** $Time \leq MaxTime$ **do**
  Discard chosen TS, and update $Obj^{current}$, $Res^{current}$, and $Rou^{current}$.
  $Res^* = Res^{current}$; $Rou^* = Rou^{current}$; $Car_t^* = Car_t^{current}$; $Obj^* = Obj^{current}$.
  **Add/Drop**[$Obj^{new}$, $Res^{new}$, $Rou^{new}$, $Car_t^{new}$].
  Select TSs.
  **if** Budget allows **then**
    **Insert**[$Obj^{new}$, $Res^{new}$, $Rou^{new}$];
  **else**
    **Erase**[$Obj^{new}$, $Res^{new}$, $Rou^{new}$, $Car_t^{new}$].
  **end if**
  Call $Metropolis(Res^{current}, Rou^{current}, Car_t^{current}, Obj^{current}, T, M)$;
  $Time = Time + M$;
  $T = \alpha \cdot T$; $M = \beta \cdot M$.
**end while**
**return** $Res^*, Rou^*, Car_t^*, Obj^*$.

The core of the SA algorithm is the *Metropolis* procedure. The Metropolis procedure, after receiving the current solution $Res^{current}, Rou^{current}, Car_t^{current}$, the temperature, $T$, and the number of metropolis loops, $M$, as inputs, simulates the annealing process at a given temperature $T$. In the Metropolis procedure, we utilize exchange and relocate neighborhoods, similar to Keskin et al. (2011), to define a new solution. We accept the "first-best-solution" in the neighborhoods. The Metropolis procedure is presented in Display 5.

**Algorithm 5** Procedure $Metropolis(Res^{current}, Rou^{current}, Car_t^{current}, Obj^{current}, T, M)$:

**while** $M > 0$ **do**
  **Relocate** & **exchange operator**$[Obj^{new},\ Res^{new},\ Rou^{new}]$.
  $\Delta Obj = Obj^{current} - Obj^{new}$.
  **if** $\Delta Obj \leq 0$ **then**
    $Rou^{current} = Rou^{new}$;$Car_t^{current} = Car_t^{new}$; $Obj^{current} = Obj^{new}$.
    **if** $Obj^{new} \leq Obj^*$ **then**
      $Rou^* = Rou^{new}$; $Car_t^* = Car_t^{new}$; and $Obj^* = Obj^{new}$.
    **end if**
  **else**
    **if** $Random < exp(-\frac{\Delta Obj}{T})$ **then**
      $Rou^{current} = Rou^{new}$; $Car_t^{current} = Car_t^{new}$; and $Obj^{current} = Obj^{new}$.
    **end if**
  **end if**
  $M = (M - 1)$.
**end while**
**return** $Obj^*,\ Res^*,\ Rou^*,\ Car_t^*$

### 2.4.2 Modification of the Heuristic for the FC Model

For the extended model, we revise DMD-MCPRP heuristic to encompass the fixed cost of TS. Specifically, the inclusion of fixed costs changes two main components of the algorithm. First of all, instead of locating the TS locations based on proximity to the starting and ending HSs in the route, we utilize a cost-based approach. We select TS locations with the smallest $c \times d_{ij}^t + F_i$ among the potential TS locations that conform to $D_{\text{limit}}$. Secondly, since **Add/Drop** most aggressively adjusts the resource consumption by changing the number of routes $Car_t$, the algorithm moves onto the modification of TS locations after the resource consumption reaches the total available budget. To improve on the selection of TS locations, we include a **Decrease TS** component that adjusts the resource consumption less aggressively by dropping one open TS at one time until the resource consumption $Res$ drops to $\mathcal{B} + av$. The rest of the algorithm, including the SA component, stays intact.

## 2.5 COMPUTATIONAL EXPERIMENTS

In order to test the proposed models and solution approaches, we design small to medium size instances from crash history data in the state of Alabama. All of the crash data in the state of Alabama since 2001 is collected by the Critical Analysis Reporting Environment (CARE)  a data analysis software package that is developed by the researchers at the University of Alabama (Steil and Parrish, 2009). To determine the effects of various factors on the performance of the heuristics as well as the coverage benefits, we design a set of experiments by varying the number of periods $|\mathcal{T}|$, the number of HSs per period $|\mathcal{N}|$, the number of depots $|\mathcal{A}|$, and the number of cars $|\mathcal{K}|$. We assume that $|\mathcal{K}|$ is positively correlated with $|\mathcal{A}|$ and $|\mathcal{I}|$ is correlated with $|\mathcal{T}|$ and $|\mathcal{N}|$. That is, if there are more depots, there should be proportionally more cars too; it is the same for $|\mathcal{I}|$ by the same token. Once the number of HSs are determined by the experimental design, we use CARE to extract the necessary HS information related to location, HS duration, and time window considerations. With this construction, our design has $2^5 = 16$ instances. The details are provided in Table 5.

Table 5: Design of experiment.

| Item | Small | Medium |
|---|---|---|
| $\|\mathcal{T}\|$ | 2 | 4 |
| $\|\mathcal{N}\|$ | 16 | 32 |
| $\|\mathcal{A}\|$ | 2 | 3 |
| $\|\mathcal{K}\|$ | $\|\mathcal{A}\| \times 2$ | $\|\mathcal{A}\| \times 3$ |
| $\|\mathcal{I}\|$ | $\|\mathcal{T}\| \times \|\mathcal{N}\| \times 1/8$ | $\|\mathcal{T}\| \times \|\mathcal{N}\| \times 1/4$ |

Based on the aforementioned design, we test all instances for

- two weight schemes $w_j^t$: high variance (1, 1.5, 2), and low variance (1, 1.1, 1.2);
- three starting strategies: **STR1**, **STR2**, and **STR3**;
- three routing cost allocation percentage levels $a$: 0.25, 0.5, and 0.75; and
- five budget levels: 20%$B$, 40%$B$, 60%$B$, 80%$B$, and 100%$B$, where $B$ is the total cost estimated when all $|\mathcal{K}|$ troopers are used for each period, and all HSs are covered on a straight-and-back basis.

In total, we run our metaheuristics for $32 \times 2 \times 3 \times 3 \times 5 = 2880$ times. We conduct all of these experiments using C++ on an Intel Core 2 Duo E8400 with 2.94GB of memory. Our proposed metaheuristics return the coverage benefit under the given budget limit. Meanwhile, since our model is a MILP, CPLEX can solve it theoretically. However, due to the complexity level of our problem and our problem size, CPLEX has difficulty in closing the optimality gap. We let CPLEX run for up to 3600 seconds. We obtain a lower bound (LB) and an upper bound. We use the best feasible solution, which is the LB, as our benchmark. We do not choose the upper bound because we do not want to mislead the results. In total, we run CPLEX for $32 \times 2 \times 5 = 320$ times.

### 2.5.1 Impact of Factor Size

When the size of the problem is small, CPLEX finds the optimal solution fairly quickly. When the size of the problem gets bigger, CPLEX takes longer to converge. In instances of bigger sizes, CPLEX runs out of memory and terminates with big optimality gaps. Our metaheuristic's runtime is similar to that of CPLEX. When the problem size is small, our metaheuristic takes less than 1 second; when the problem size gets bigger, our metaheuristic takes longer but still under 1 minute. However, the solution quality of our approach moves in the opposite direction as that of CPLEX. When CPLEX returns no or small optimality gaps, our solution performs equivalently to, if not worse than, CPLEX. But when CPLEX struggles to solve instances of bigger sizes, our solution approach outperforms CPLEX by a large extent.

### 2.5.2 Experiment for DMD-MCPRP

After obtaining the coverage objective from the heuristic and the LB from CPLEX, we evaluate our solution approach by examining the gap: $(Objective - LB)/LB$. If the gap is positive, our heuristic finds a better solution than the best feasible solution that CPLEX is able to find within the given runtime. However, it is also possible that the LB of CPLEX is better than our heuristic, i.e., the gap is negative. We report both the average and maximum gap, in short "Avg." and "Max." in Table 6.

Table 6: Comparison of metaheuristic and CPLEX.

| | $w_i^t$ =(1, 1.1, 1.2) | | | | | | | | $w_i^t$ =(1, 1.5, 2) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. (%) | | | | Max. (%) | | | | Avg. (%) | | | | Max. (%) | | | |
| a=0.25 | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best |
| 100%$\mathcal{B}$ | 2.3 | 0.8 | 1.4 | 2.6 | 22.2 | 20.9 | 18.9 | 22.2 | 0.6 | -0.4 | 0.0 | 1.2 | 22.8 | 19.7 | 19.6 | 22.8 |
| 80%$\mathcal{B}$ | 1.7 | 0.2 | 0.8 | 2.0 | 18.0 | 14.8 | 16.0 | 18.0 | 1.1 | 0.0 | 0.4 | 1.6 | 23.4 | 20.2 | 20.2 | 23.4 |
| 60%$\mathcal{B}$ | 1.6 | 0.1 | 0.8 | 2.0 | 20.5 | 19.2 | 17.3 | 20.5 | 0.6 | -0.4 | 0.0 | 1.2 | 13.6 | 11.1 | 10.4 | 13.6 |
| 40%$\mathcal{B}$ | 0.4 | -1.6 | -0.7 | 0.8 | 21.5 | 19.9 | 18.1 | 21.5 | 0.9 | -0.5 | -0.2 | 1.4 | 21.3 | 20.0 | 19.3 | 21.3 |
| 20%$\mathcal{B}$ | -1.4 | -5.0 | -2.8 | -0.6 | 55.9 | 53.3 | 55.9 | 55.9 | -0.2 | -3.7 | -1.7 | 0.9 | 56.1 | 56.1 | 51.5 | 56.1 |
| a=0.5 | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best |
| 100%$\mathcal{B}$ | 2.3 | 0.8 | 1.4 | 2.6 | 22.2 | 20.9 | 18.9 | 22.2 | 0.6 | -0.4 | 0.0 | 1.2 | 22.8 | 19.7 | 19.6 | 22.8 |
| 80%$\mathcal{B}$ | 1.7 | 0.2 | 0.8 | 2.0 | 18.0 | 14.8 | 16.0 | 18.0 | 1.1 | 0.0 | 0.4 | 1.6 | 23.4 | 20.2 | 20.2 | 23.4 |
| 60%$\mathcal{B}$ | 1.2 | -0.5 | 0.5 | 1.7 | 20.5 | 19.2 | 17.3 | 20.5 | 0.6 | -0.6 | -0.3 | 1.0 | 13.6 | 11.1 | 10.4 | 13.6 |
| 40%$\mathcal{B}$ | -1.6 | -2.2 | -1.4 | -0.2 | 21.5 | 19.9 | 18.1 | 21.5 | -0.5 | -1.0 | -0.3 | 0.6 | 21.3 | 20.0 | 19.3 | 21.3 |
| 20%$\mathcal{B}$ | -3.7 | -4.8 | -4.6 | -2.2 | 57.0 | 57.9 | 58.5 | 58.5 | -3.1 | -2.9 | -2.5 | -0.4 | 60.0 | 53.7 | 53.5 | 60.0 |
| a=0.75 | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best |
| 100%$\mathcal{B}$ | 1.6 | -0.1 | 0.8 | 2.0 | 22.2 | 20.9 | 18.9 | 22.2 | -0.1 | -1.1 | -0.8 | 0.5 | 22.8 | 19.7 | 19.6 | 22.8 |
| 80%$\mathcal{B}$ | -0.5 | -1.2 | -0.4 | 0.6 | 18.0 | 14.8 | 16.0 | 18.0 | -0.6 | -0.8 | -0.4 | 0.6 | 23.4 | 20.2 | 20.2 | 23.4 |
| 60%$\mathcal{B}$ | -1.9 | -1.5 | -0.8 | 0.2 | 17.7 | 16.0 | 16.5 | 17.7 | -2.3 | -1.4 | -1.7 | -0.3 | 13.1 | 12.1 | 14.1 | 14.1 |
| 40%$\mathcal{B}$ | -4.6 | -3.7 | -3.6 | -2.4 | 19.4 | 18.8 | 18.5 | 19.4 | -4.5 | -2.0 | -2.5 | -1.0 | 15.5 | 18.4 | 20.0 | 20.0 |
| 20%$\mathcal{B}$ | -6.7 | -6.6 | -5.5 | -3.5 | 54.8 | 57.9 | 56.7 | 57.9 | -3.8 | -3.2 | -3.7 | -1.1 | 52.1 | 52.5 | 56.7 | 56.7 |

In Table 6, if we compare different budget levels, there is a general trend: as the budgets become tighter and tighter, the average gaps become slightly worse. The only exception is when $a = 0.25$ with low variance weights where budgets 80%$\mathcal{B}$ and 60%$\mathcal{B}$ have equivalent average gaps. At 100%$\mathcal{B}$, 80%$\mathcal{B}$, and 60%$\mathcal{B}$, with best starting strategy, our heuristic outperforms the LB returned by CPLEX. Therefore, if there is enough budget, our metaheuristic displays a dominating advantage over CPLEX. On the other hand, if the budget is tight, this dominance is only compromised slightly. The inclusion of the budget limit allows us to conduct the benefit-cost tradeoff analysis; its result shows how much a change in the budget will affect the patrol effectiveness.

Next, we compare different starting strategies. Both with lower and higher variance weights, **Str1** has the best gap for $a = 0.25$ with all budget levels and for $a = 0.50$ with most budget levels. However, for $a = 0.75$, there is no consistent result with respect to which one is the best. For instance, with lower variance weights, **Str3** has the best gap for most budget levels and with high variance weights, **Str2** has the best gap for budget levels 60%$\mathcal{B}$, 40%$\mathcal{B}$, and 20%$\mathcal{B}$. Because of the lack in consistency, it is recommended to use the all of the starting strategies and pick the best one. Since the heuristic is running fast, this does not create additional problems.

Third, different route cost allocations to factor $a$ do not affect the coverage benefit. We conduct an indicator variable analysis by taking values in columns "Best" for average and best performances as responses and introducing two indicators for three levels of $a$. The statistical $p$ values of both indicator variables are 0.880 and 0.876, respectively. Therefore, the proposed metaheuristic is robust. In the first two rows with $a = 0.25$ and those with $a = 0.5$, the results are exactly the same. Regardless of how much budget is allocated to gas consumption in the beginning of the algorithm, the inherent **Add/Drop** component adjusts the number of cars very effectively. The robustness is substantiated further, if we compare different $w_i^t$. Low-variance weights and high-variance weights have quite similar coverage benefits and similar heuristic performances. The robustness of our approach is critical when decision makers have different perceptions of different crash types and assign different weights to them.

Overall, the best performances by the heuristic outperform those by CPLEX. The largest improvement reaches up to 60.0%, that is, optimistically speaking, our method provides state troopers with 60.0% more coverage than the commercial software does. All of the five factors have positive impact on the objective. Especially, the positive relation between the number of TSs and the coverage objective forms the root cause of the necessity to incorporate TSs in the patrol routes of state troopers.

We also report the average runtime (in seconds) of 32 instances in Table 7. Our metaheuristic runtime is averaged to be less than one second, while CPLEX takes an hour to obtain a LB. The solution time is very critical, especially when state troopers need to respond to accidents in a timely manner. Therefore, our solution approach is more favorable.

Table 7: Average runtime of metaheuristic.

| | $w_i^t$ =(1, 1.1, 1.2) | | | $w_i^t$ =(1, 1.5, 2) | | |
|---|---|---|---|---|---|---|
| a=0.25 | Str1 | Str2 | Str3 | Str1 | Str2 | Str3 |
| 100%$\mathcal{B}$ | 0.38 | 0.38 | 0.37 | 0.41 | 0.38 | 0.40 |
| 80%$\mathcal{B}$ | 0.38 | 0.37 | 0.37 | 0.41 | 0.37 | 0.40 |
| 60%$\mathcal{B}$ | 0.60 | 0.37 | 0.37 | 0.41 | 0.37 | 0.40 |
| 40%$\mathcal{B}$ | 0.36 | 0.36 | 0.35 | 0.40 | 0.39 | 0.38 |
| 20%$\mathcal{B}$ | 0.30 | 0.30 | 0.66 | 0.31 | 0.29 | 0.46 |
| a=0.50 | Str1 | Str2 | Str3 | Str1 | Str2 | Str3 |
| 100%$\mathcal{B}$ | 0.40 | 0.37 | 0.38 | 0.42 | 0.36 | 0.41 |
| 80%$\mathcal{B}$ | 0.39 | 0.37 | 0.38 | 0.42 | 0.37 | 0.42 |
| 60%$\mathcal{B}$ | 0.39 | 0.37 | 0.38 | 0.41 | 0.37 | 0.41 |
| 40%$\mathcal{B}$ | 0.38 | 0.36 | 0.37 | 0.39 | 0.36 | 0.40 |
| 20%$\mathcal{B}$ | 0.31 | 0.29 | 0.30 | 0.29 | 0.27 | 0.32 |
| a=0.75 | Str1 | Str2 | Str3 | Str1 | Str2 | Str3 |
| 100%$\mathcal{B}$ | 0.39 | 0.37 | 0.38 | 0.40 | 0.36 | 0.39 |
| 80%$\mathcal{B}$ | 0.39 | 0.37 | 0.38 | 0.40 | 0.36 | 0.40 |
| 60%$\mathcal{B}$ | 0.40 | 0.38 | 0.39 | 0.41 | 0.36 | 0.38 |
| 40%$\mathcal{B}$ | 0.39 | 0.36 | 0.38 | 0.37 | 0.36 | 0.38 |
| 20%$\mathcal{B}$ | 0.29 | 0.27 | 0.33 | 0.29 | 0.28 | 0.28 |

### 2.5.3 Experiment for the Extended Model

Next, we investigate the performance of the revised metaheuristic to solve the extended model with $F_i$. Keeping all other parameters the same, we test our algorithm with identical $F_i = \{2, 8\}$ \$/TS/period since each TS is charged the same. If $F_i$ is TS dependent, our algorithm is generic enough to handle as well. Since weights (1, 1.1, 1.2) and (1, 1.5, 2) have very similar results, we only report the results of one weight scheme - (1, 1.1, 1.2) to avoid redundancy. The reported items are the gaps compared with CPLEX, shown in Table 8.

Table 8: Comparison of revised metaheuristic and CPLEX for $w_i^t$ =(1, 1.1, 1.2) with different fixed costs.

| | $F_i = 2$ | | | | | | | | $F_i = 8$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. (%) | | | | Max. (%) | | | | Avg. (%) | | | | Max. (%) | | | |
| a=0.25 | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best |
| 100%$\mathcal{B}$ | 0.6 | -0.8 | -0.2 | 1.0 | 23.6 | 22.3 | 20.4 | 23.6 | -0.6 | -2.0 | -1.4 | -0.2 | 27.9 | 26.5 | 24.5 | 27.9 |
| 80%$\mathcal{B}$ | 0.2 | -1.2 | -0.6 | 0.6 | 13.5 | 14.1 | 13.2 | 14.1 | 0.4 | -1.0 | -0.4 | 0.8 | 15.4 | 14.2 | 12.4 | 15.4 |
| 60%$\mathcal{B}$ | 0.1 | -1.3 | -0.7 | 0.5 | 16.1 | 14.9 | 13.0 | 16.1 | -0.2 | -1.7 | -1.1 | 0.1 | 28.1 | 26.8 | 24.7 | 28.1 |
| 40%$\mathcal{B}$ | 0.7 | -1.3 | -0.2 | 1.2 | 21.6 | 20.3 | 18.4 | 21.6 | -0.3 | -1.9 | -0.9 | 0.5 | 16.6 | 17.2 | 16.3 | 17.2 |
| 20%$\mathcal{B}$ | -2.5 | -4.4 | -3.7 | -1.1 | 34.8 | 28.7 | 36.4 | 36.4 | -4.2 | -6.4 | -4.6 | -1.0 | 71.4 | 66.0 | 75.5 | 75.5 |
| a=0.5 | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best |
| 100%$\mathcal{B}$ | 0.6 | -0.8 | -0.2 | 1.0 | 23.6 | 22.3 | 20.4 | 23.6 | -0.6 | -2.0 | -1.4 | -0.2 | 27.9 | 26.5 | 24.5 | 27.9 |
| 80%$\mathcal{B}$ | 0.2 | -1.2 | -0.6 | 0.6 | 13.5 | 14.1 | 13.2 | 14.1 | 0.4 | -1.0 | -0.4 | 0.8 | 15.4 | 14.2 | 12.4 | 15.4 |
| 60%$\mathcal{B}$ | -0.3 | -1.9 | -1.0 | 0.2 | 16.1 | 14.9 | 13.0 | 16.1 | -0.7 | -2.3 | -1.4 | -0.2 | 28.1 | 26.8 | 24.7 | 28.1 |
| 40%$\mathcal{B}$ | -1.3 | -2.2 | -1.3 | 0.1 | 21.6 | 20.3 | 18.4 | 21.6 | -2.0 | -2.6 | -1.8 | -0.6 | 16.6 | 17.2 | 16.3 | 17.2 |
| 20%$\mathcal{B}$ | -5.4 | -4.4 | -5.1 | -2.6 | 30.6 | 37.0 | 34.6 | 37.0 | -3.6 | -3.2 | -3.1 | -0.1 | 80.1 | 89.9 | 76.9 | 89.9 |
| a=0.75 | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best | Str1 | Str2 | Str3 | Best |
| 100%$\mathcal{B}$ | 0.0 | -1.7 | -0.8 | 0.4 | 23.6 | 22.3 | 20.4 | 23.6 | -1.2 | -2.9 | -2.0 | -0.8 | 27.9 | 26.5 | 24.5 | 27.9 |
| 80%$\mathcal{B}$ | -2.0 | -2.6 | -1.8 | -0.8 | 13.5 | 14.1 | 13.2 | 14.1 | -1.8 | -2.4 | -1.7 | -0.6 | 15.4 | 14.2 | 12.4 | 15.4 |
| 60%$\mathcal{B}$ | -3.3 | -2.9 | -2.2 | -1.3 | 13.4 | 11.8 | 12.9 | 13.4 | -3.7 | -3.3 | -2.7 | -1.7 | 25.1 | 23.4 | 23.7 | 25.1 |
| 40%$\mathcal{B}$ | -4.5 | -3.5 | -3.7 | -2.3 | 18.5 | 16.5 | 18.2 | 18.5 | -4.9 | -4.3 | -4.4 | -2.9 | 16.8 | 13.9 | 14.6 | 16.8 |
| 20%$\mathcal{B}$ | -6.0 | -6.6 | -5.8 | -3.0 | 28.1 | 27.5 | 34.9 | 34.9 | -5.1 | -5.7 | -3.4 | -1.6 | 86.8 | 80.4 | 79.5 | 86.8 |

If we compare different starting strategies, budget levels, and $a$ levels, we get similar results as in the previous subsection. However, if we compare the results with the fixed cost and those without the fixed cost, additional insights can be drawn. When $F_i = 0$, state troopers are more spread out with respect to where they start and stop; when $F_i > 0$, state troopers tend to share the starting or stopping places in order to save money on paying for the fixed cost of TS. The more tight the budget is and the higher the fixed cost of TS is, the more obvious this phenomenon is. It can be projected that if the budget is really tight and the fixed cost of TS is high enough, all state troopers will share only one TS each period, which becomes a single depot problem.

### 2.5.4 Comparison of Performances between MCPRP and DMD-MCPRP

In addition to comparing with CPLEX, we also benchmark on work by Keskin et al. (2011) since DMD-MCPRP is an extension of MCPRP. Other than the objective, Keskin et al. (2011) also introduced two performance measures to evaluate the proposed coverage plan. They are “Percentage of Hot Spots Covered (HS%)” and “Percentage of Coverage Length (TW%)”. For the sake of completeness, we present the definitions as follows:

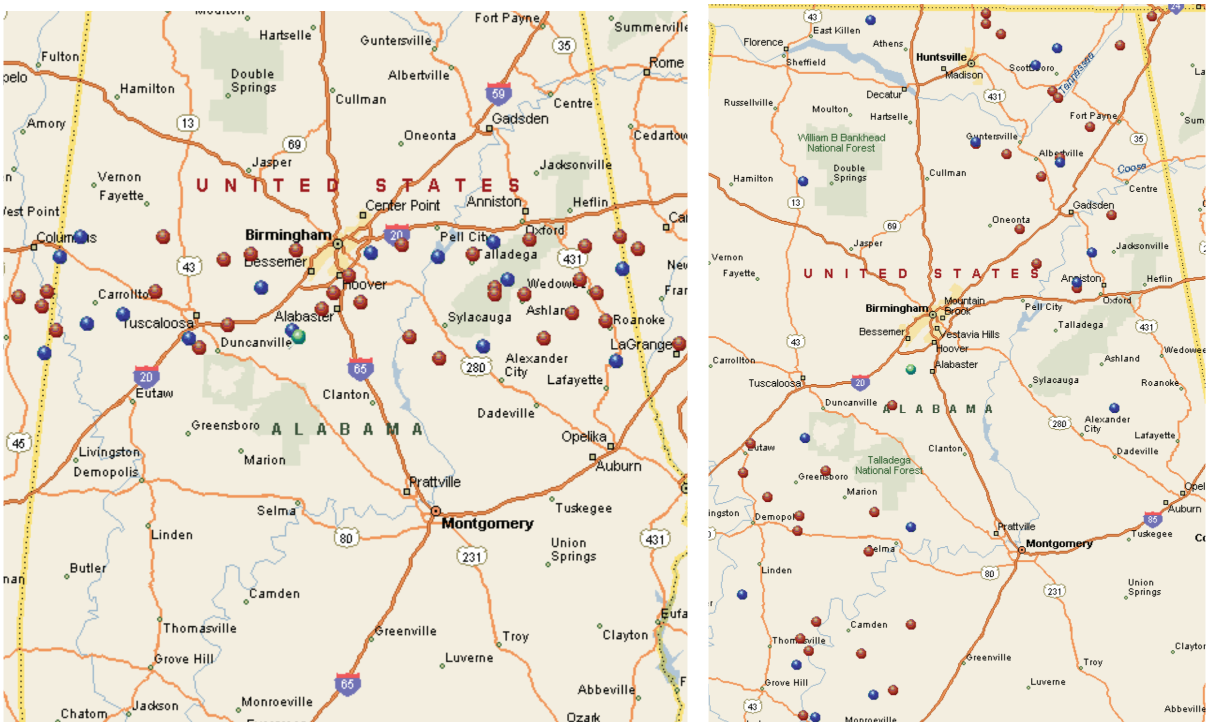
**HS%:** This performance measure calculates, among all of the hot spots, the percentage covered as a result: $HS\% = \frac{\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}_t}\sum_{k\in\mathcal{K}} y_{ik}^t}{|\mathcal{T}|\times|\mathcal{N}|}$, where the numerator represents the total number of visited hot spots.

**TW%:** This performance measure calculates the percentage of total available time serviced: $TW\% = \frac{\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}_t}\sum_{k\in\mathcal{K}}(f_{ik}^t - s_{ik}^t)}{\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}_t}(l_i^t - e_i^t)}$. In this measure, the numerator is the service time returned, and the denominator is the total time window length.

We compare objectives, HW%, and TW% of DMD-MCPRP with those of MCPRP. Since MCPRP does not have a budget limit, it is only compared with DMD-MCPRP without the fixed cost when the budget is $100\%\mathcal{B}$. The best objectives of all $a$ and all starting strategies of DMD-MCPRP are compared with objectives of MCPRP returned by local search. Based on the previous design of experiment, we set the number of TSs $|\mathcal{I}|$ equal to the product of $|\mathcal{T}|$ and $|\mathcal{N}|$, and thus $|\mathcal{I}|$ is not a factor in the new design. Totally, there are 16 instances for comparisons. Since our solution is robust with respect to different weight schemes, we focus on the low weight (1, 1.1, 1.2).

In order to better demonstrate the differences of DMD-MCPRP and MCPRPR, we randomly generate HSs based on two different distributions, which are plotted in Figure 8. The red dots represent HSs, the blue dots represent TSs including depots, the green dot represents the single depot. In Part (a), HSs are uniformly distributed within one cluster; and in Part (b), HSs are uniformly distributed within two clusters. The single depot is located in the center, and it is close to HSs in Part (a) but far from HSs in Part (b). The multiple depots are located within the cluster(s), and they are close to HSs in both parts.

We first present the results for HSs of uniform distribution in Table 9. It includes the following columns: "Ins" (instance), "TotTm" (total weighted time available), "TotHS" (total number of HSs available), "CovTm" (covered objective), "TW%", "HS%", and "Imp" (improvement in the objective in percentage). To compare objectives of MCPRP and DMD-MCPRP when $D_{\text{limit}} = 20$, we calculate the improvements in column "Imp" as $\frac{\text{"CovTm" of "DMD-MCPRP-20"} - \text{"CovTm" of MCPRP}}{\text{"CovTm" of MCPRP}} \times 100\%$. The results confirm our intuition that




(a) HSs of uniform distribution (b) HSs of clustered distribution

Figure 8: HSs of uniform distribution and clustered distribution.

DMD-MCPRP outperforms MCPRP. In column "Imp", the worst performance of DMD-MCPRP among all the instances is a tie with MCPRP in instance 1. DMD-MCPRP yields an improvement as high as 6% in instances 11 and 16. The improvement is attributed to the dynamic selection of a TS, with state troopers starting at a TS closer to HSs than the central depot and stopping at a TS closer to HSs in the next period. Meanwhile, we report the average values at the bottom of the table, referred to as "Avg.". DMD-MCPRP, on average, has 3.2% more time coverage benefits than MCPRP. Even though this percentage may seem low, in practice this translates to almost two extra hours of effective coverage. Improvement at such scale helps state troopers increase their patrol effectiveness.

Table 9: Comparison of performances between MCPRP and DMD-MCPRP with uniform distribution.

| | | | MCPRP | | | DMD-MCPRP-20 | | | | DMD-MCPRP-240 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ins | TotTm | TotHS | CovTm | TW% | HS% | CovTm | TW% | HS% | Imp | CovTm | TW% | HS% | Imp |
| 1 | 2826.6 | 32 | 2195.9 | 78% | 88% | 2195.9 | 78% | 88% | 0% | 2201.7 | 78% | 84% | 0% |
| 2 | 5769.2 | 64 | 4702.7 | 82% | 95% | 4738.1 | 82% | 94% | 1% | 4784.6 | 83% | 92% | 2% |
| 3 | 2976.5 | 32 | 2768.8 | 93% | 100% | 2800.3 | 94% | 97% | 1% | 2926 | 98% | 100% | 6% |
| 4 | 5015.8 | 64 | 4728 | 94% | 100% | 4857 | 95% | 100% | 3% | 4836.6 | 94% | 100% | 2% |
| 5 | 2815.5 | 32 | 2524.5 | 90% | 91% | 2567.3 | 91% | 97% | 2% | 2567.3 | 91% | 97% | 2% |
| 6 | 5689.7 | 64 | 5080.8 | 89% | 98% | 5345 | 93% | 100% | 5% | 5317.6 | 93% | 100% | 5% |
| 7 | 2829.3 | 32 | 2687.4 | 95% | 97% | 2810.7 | 99% | 100% | 5% | 2814.9 | 99% | 100% | 5% |
| 8 | 6058.7 | 64 | 5790.7 | 96% | 100% | 5962.5 | 98% | 100% | 3% | 5967.8 | 98% | 100% | 3% |
| 9 | 5469.5 | 64 | 2701.8 | 49% | 63% | 2819.9 | 49% | 70% | 4% | 2861.2 | 49% | 70% | 6% |
| 10 | 11588.9 | 128 | 5665.9 | 49% | 67% | 5848 | 50% | 63% | 3% | 5895.3 | 50% | 63% | 4% |
| 11 | 5559.7 | 64 | 3752.6 | 67% | 94% | 3994.7 | 72% | 89% | 6% | 3988.1 | 72% | 89% | 6% |
| 12 | 11802.4 | 128 | 8120.1 | 69% | 91% | 8313.7 | 69% | 88% | 2% | 8286.7 | 70% | 88% | 2% |
| 13 | 5439.2 | 64 | 3658.9 | 67% | 89% | 3847.7 | 69% | 81% | 5% | 3898.8 | 72% | 81% | 7% |
| 14 | 10391 | 128 | 7375.2 | 71% | 86% | 7492.1 | 71% | 88% | 2% | 7434.5 | 71% | 88% | 1% |
| 15 | 5569.9 | 64 | 4678.8 | 84% | 100% | 4811.1 | 85% | 98% | 3% | 4820.9 | 85% | 98% | 3% |
| 16 | 11656.5 | 128 | 9801.8 | 84% | 98% | 10386 | 88% | 97% | 6% | 10556.9 | 90% | 97% | 8% |
| Avg. | | | | 79% | 91% | | 80% | 91% | 3.2% | | 81% | 91% | 3.8% |

The TW% performances of both MCPRPR and DMD-MCPRP are consistent with objectives, as they have the same denominators. On average, DMD-MCPRP returns 80% and MCPRP returns 79% TW% coverage. Thus, with DMD-MCPRP, state troopers stay at HSs longer. With respect to the HS%, there is no relationship between DMD-MCPRP and MCPRP. On average, DMD-MCPRP returns 90% and MCPRP returns 91% HS% coverage. Thus, with MCPRP, state troopers move more often from one HS to another due to the elapse of the time window.

We also vary $D_{\text{limit}}$ to show its impact on the performance of our solution approach. We set $D_{\text{limit}}$ to a higher value of 240 minutes and rerun the same 16 instances. The results are shown in the column "DMD-MCPRP-240" in Table 9. "DMD-MCPRP-240" further improves "Imp" from 3.2% by "DMD-MCPRP-20" to 3.8% on average.

We then present the results for HSs of clustered distribution in Table 10. It has the same columns as the previous table does. The findings are similar to those in the previous distribution. The differences lie in the magnitudes of improvements over MCPRP. On average, DMD-MCPRP improves MCPRP by 9.6% when $D_{\text{limit}} = 20$ and MCPRP by 11.7% when $D_{\text{limit}} = 240$. TW% and HS% are improved from 60% and 75% to 64% and 78%, respectively, when $D_{\text{limit}} = 20$, and further improved to 67% and 78% when $D_{\text{limit}} = 240$.

Table 10: Comparison of performances between MCPRP and DMD-MCPRP with clustered distribution.

| | | | MCPRP | | | DMD-MCPRP-20 | | | | DMD-MCPRP-240 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ins | TotTm | TotHS | CovTm | TW% | HS% | CovTm | TW% | HS% | Imp | CovTm | TW% | HS% | Imp |
| 1 | 2826.6 | 32 | 1492.6 | 53% | 69% | 1705.1 | 60% | 72% | 14% | 1670.8 | 59% | 69% | 12% |
| 2 | 5769.2 | 64 | 3345.5 | 58% | 73% | 3520.1 | 61% | 81% | 5% | 3624.8 | 63% | 81% | 8% |
| 3 | 2976.5 | 32 | 2126.6 | 71% | 94% | 2531.5 | 73% | 88% | 19% | 2637.5 | 89% | 91% | 24% |
| 4 | 5015.8 | 64 | 3850.8 | 77% | 86% | 4169.2 | 83% | 95% | 8% | 4133.9 | 82% | 94% | 7% |
| 5 | 2815.5 | 32 | 2102.2 | 75% | 81% | 2197.7 | 76% | 84% | 5% | 2214.8 | 79% | 81% | 5% |
| 6 | 5689.7 | 64 | 4440.4 | 78% | 92% | 4658.7 | 82% | 94% | 5% | 4713.5 | 83% | 94% | 6% |
| 7 | 2829.3 | 32 | 2184.8 | 77% | 88% | 2525.6 | 85% | 94% | 16% | 2555.3 | 90% | 94% | 17% |
| 8 | 6058.7 | 64 | 4785 | 79% | 91% | 5551.5 | 91% | 100% | 16% | 5607.5 | 93% | 98% | 17% |
| 9 | 5469.5 | 64 | 1842.3 | 34% | 44% | 1948.2 | 36% | 47% | 6% | 2038.9 | 37% | 50% | 11% |
| 10 | 11588.9 | 128 | 3797.8 | 33% | 46% | 3986.1 | 33% | 45% | 5% | 4107.6 | 35% | 48% | 8% |
| 11 | 5559.7 | 64 | 2916.7 | 52% | 67% | 3036.7 | 52% | 69% | 4% | 3091.9 | 56% | 69% | 6% |
| 12 | 11802.4 | 128 | 5490.2 | 47% | 67% | 6191.7 | 52% | 66% | 13% | 6273 | 53% | 70% | 14% |
| 13 | 5439.2 | 64 | 2565.3 | 47% | 64% | 2982 | 52% | 70% | 16% | 3047.6 | 56% | 69% | 19% |
| 14 | 10391 | 128 | 5145.5 | 50% | 66% | 5626.9 | 54% | 65% | 9% | 5748.4 | 55% | 66% | 12% |
| 15 | 5569.9 | 64 | 3832.6 | 69% | 89% | 4029.5 | 71% | 91% | 5% | 4043.5 | 73% | 89% | 6% |
| 16 | 11656.5 | 128 | 7683.9 | 66% | 84% | 8298.9 | 71% | 89% | 8% | 8761.1 | 75% | 89% | 14% |
| Avg. | | | | 60% | 75% | | 64% | 78% | 9.6% | | 67% | 78% | 11.7% |

Therefore, for both uniformly distributed and clustered distributed HSs, DMD-MCPRP outperforms MCPRP. DMD-MCPRP is moderately better than MCPRP for uniformly distributed HSs, and DMD-MCPRP is significantly better than MCPRP for clustered distributed HSs.

## 2.6 CONCLUSIONS

In conclusion, in order to improve the efficiency of state trooper patrolling, we allow for dynamically changing patrol routes and starting and stopping locations. To account for the dynamism, we develop a new dynamic, multi-depot, location-routing model, extending MCPRP in the literature. Gaining insights from solutions of LRP, we decompose this problem into multi-depot MCPRP and facility location, and then solve them in an iterative way with custom built heuristics. We test the model and solution approach for the situations without and with a fixed cost of TS, and compare it with the LB of CPLEX. We also compare the time and hot spot coverage performances of this model with the single depot MCPRP, and significant improvements are found in the objectives.

There are several extensions for future research. One possible extension is to include dynamic travel times with the real time traffic conditions. Another extension, in addition to covering predetermined HSs, is to consider on-call responses of state troopers.

## REFERENCES


Albareda-Sambola, M., E. Fernández, G. Laporte. 2007. Heuristic and lower bound for a stochastic location-routing problem. *European Journal of Operational Research* **179**(3) 940–955.

Alçada-Almeida, L., L. Tralhão, L. Santos, J. Coutinho-Rodrigues. 2009. A multiobjective approach to locate emergency shelters and identify evacuation routes in urban areas. *Geographical Analysis* **41**(1) 9–29.

Alumur, S., B. Y. Kara. 2007. A new model for the hazardous waste location-routing problem. *Computers and Operations Research* **34**(5) 1406–1423.

Balakrishnan, A., J. E. Ward, R. T. Wong. 1987. Integrated facility location and vehicle routing models: Recent work and future prospects. *American Journal of Mathematical and Management Sciences* **7**(1) 35–61.

Bérubé, J.F., M. Gendreau, J. Y. Potvin. 2009. An exact-constraint method for bi-objective combinatorial optimization problems: Application to the Traveling Salesman Problem with Profits. *European Journal of Operational Research* **194**(1) 39–50.

Blincoe, L., A. Seay, E. Zaloshnja, T. Miller, E. Romano, S. Luchter, R. Spicer. 2002. The economic impact of motor vehicle crashes, 2000. Tech. rep., Crash Statistics DOT HS 809 446, NHTSA. www-nrd.nhtsa.dot.gov.

Boussier, S., D. Feillet, M. Gendreau. 2007. An exact algorithm for team orienteering problems. *4OR: A Quarterly Journal of Operations Research* **5**(3) 211–230.

Caballero, R., M. González, F. M. Guerrero, J. Molina, C. Paralera. 2007. Solving a multi-objective location routing problem with a metaheuristic based on tabu search. Application to a real case in Andalusia. *European Journal of Operational Research* **177**(3) 1751–1763.

Chankong, V., Y. Y. Haimes. 1983. *Multiobjective Decision Making: Theory and Methodology*, vol.8. North-Holland.

Hansen, P. H., B. Hegedahl, S. Hjortkjaer, B. Obel. 1994. A heuristic solution to the warehouse location-routing problem. *European Journal of Operational Research* **76**(1) 111–127.

Keskin, B. B., S. R. Li, D. Steil, S. Spiller. 2011. Analysis of an integrated maximum covering and patrol routing problem. *Transportation Research Part E: Logistics and Transportation Review* **48**(1) 215-232.

Kirkpatrick, S., C. D. Gelatt Jr., M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* **220** 671-680.

Labbé, M., I. Rodríguez-Martin, J. J. Salazar-González. 2004. A branch-and-cut algorithm for the plant-cycle location problem. *Journal of the Operational Research Society* **55**(5) 513–520.

Laporte, G. 1988. Location-routing problems. B. L. Golden, A. A. Assad, eds., *Vehicle Routing: Methods and Studies*. Noth-Holland, Amsterdam, 163–198.

Laporte, G., Y. Nobert, D. Arpin. 1986. An exact algorithm for solving a capacitated location-routing problem. *Annals of Operations Research* **6**(9) 291–310.

Laumanns, M., L. Thiele, E. Zitzler. 2005. An adaptive scheme to generate the pareto front based on the $\epsilon$-constraint method. *Practical Approaches to Multi-Objective Optimization*.

Laumanns, M., L. Thiele, E. Zitzler. 2006. An efficient, adaptive parameter variation scheme for metaheuristics based on the $\epsilon$-constraint method. *European Journal of Operational Research* **169**(3) 932–942.

Lou, Y., Y. Yin, S. Lawphongpanich. 2011. Freeway service patrol deployment planning for incident management and congestion mitigation. *Transportation Research Part C: Emerging Technologies* **19**(2) 283-295.

Mavrotas, G. 2009. Effective implementation of the $\epsilon$-constraint method in Multi-Objective Mathematical Programming problems. *Applied Mathematics and Computation* **213**(2) 455–465.

Melechovskỳ, J., C. Prins, R. W. Calvo. 2005. A metaheuristic to solve a location-routing problem with non-linear costs. *Journal of Heuristics* **11**(5) 375–391.

Miettinen, K. 1999. *Nonlinear Multiobjective Optimization*, vol.12. Springer.

Min, H., V. Jayaraman, R. Srivastava. 1998. Combined location-routing problems: A synthesis and future research directions. *European Journal of Operational Research* **108**(1) 1–15.

Montemanni, R., L. M. Gambardella. 2009. Ant colony system for team orienteering problems with time windows. *Foundations of Computing and Decision Sciences* **34**(4) 287–306.

Nagy, G., S. Salhi. 1996. Nested heuristic methods for the location-routeing problem. *Journal of the Operational Research Society* **47**(9) 1166–1174.

Nagy, G., S. Salhi. 2007. Location-routing: Issues, models and methods. *European Journal of Operational Research* **177**(2) 649–672.

Perl, J., M. S. Daskin. 1985. A warehouse location-routing problem. *Transportation Research Part B: Methodological* **19**(5) 381–396.

Sait, S. M., H. Youssef. 1999. *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems*. IEEE Computer Society, Los Alamitos, CA.

Salhi, S., G. K. Rand. 1989. The effect of ignoring routes when locating depots. *European Journal of Operational Research* **39**(2) 150–156.

Semet, F., E. Taillard. 1993. Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research* **41**(4) 469–488.

Steil, D., A. Parrish. 2009. Hit: A gis-based hotspot identification taxonomy. *International Journal of Computers and Their Applications* **16**(2) 1–10.

Tricoire, F., M. Romauch, K. F. Doerner, R. F. Hartl. 2010. Heuristics for the multi-period orienteering problem with multiple time windows. *Computers and Operations Research* **37**(2) 351–367.

Tsiligirides, T. 1984. Heuristic methods applied to orienteering. *Journal of the Operational Research Society* **35**(9) 797–809.

Vansteenwegen, P., W. Souffriau, G. Vanden Berghe, D. Van Oudheusden. 2009. Iterated local search for the team orienteering problem with time windows. *Computers and Operations Research* **36**(12) 3281–3290.

Willemse, E. J., J. W. Joubert. 2012. Applying minmax k postmen problems to the routing of security guards. *Journal of the Operational Research Society* **63**(2) 245–260.

Wu, T. H., C. Low, J. W. Bai. 2002. Heuristic solutions to multi-depot location-routing problems. *Computers and Operations Research* **29**(10) 1393–1415.

APPENDIX

Prove Observation 1 is true by induction for one car case.

We denote $S$ as an optimal solution by selecting the optimal TS according to Observation 1, and denote $S'$ as the revised solution when a different TS′ is chosen. Let $v(S)$ and $v(S)'$ denote the objective function values of $S$ and $S'$, respectively.

**Step one**: prove Observation 1 is true when there are 2 periods.

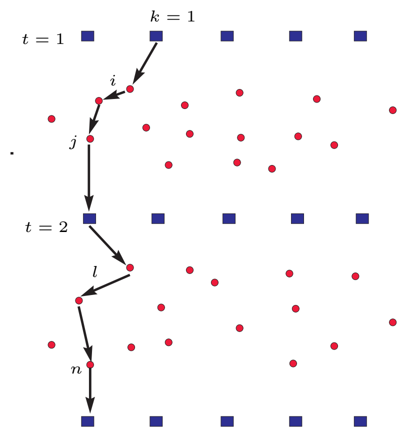



Figure 9: Step one

At the beginning of $t = 1$, we know the optimal route information with respect to HS coverage. $i$ is the first covered HS and the start time is $s^1_{i1}$. Observation 1 picks the closest TS. Assume another TS' is picked. Then, the new start time is $s^{1'}_{i1}$, and the rest of the route is the same.

$$
\begin{aligned}
& v(S) - v(S') \\
& = -s^1_{i1} + s^{1'}_{i1} \\
& \geq 0, \text{due to } d^1_{TS,i} < d^1_{TS',i},\ s^1_{i1} = \max\{d^1_{TS,i}, e^1_i\},\ s^{1'}_{i1} = \max\{d^1_{TS',i}, e^1_i\}
\end{aligned}
$$

If a TS other than the closest TS is chosen, the objective is no better. Thus, by proof of contradiction, we pick the closest TS for the beginning of $t = 1$.

When $t = 2$, we know the optimal route information for both periods 1 and 2. $j$ is the last covered HS on the route when $t = 1$, and the stop time is $f^1_{j1}$; $l$ is the first covered HS on the route when $t = 2$, and the start time is $s^2_{l1}$. Observation 1 picks the closest TS to these end points. Assume another TS' is picked. Then, the new finish time and start time are $f_{j1}^{1'}$ and $s_{l1}^{2'}$, and the rest of the two routes is the same.

$$\begin{aligned}&v(S) - v(S')\\&= f_{j1}^{1} - f_{j1}^{1'} - s_{l1}^{2} + s_{l1}^{2'}\\&\geq 0, \text{due to } d_{j,TS}^{1} < d_{j,TS'}^{1},\ d_{TS,l}^{2} < d_{TS',l}^{2},\ f_{j1}^{1} = \min\{480 - d_{j,TS}^{1}, l_j^1\},\ f_{j1}^{1'} = \min\{480 - d_{j,TS'}^{1}, l_j^1\}\end{aligned}$$

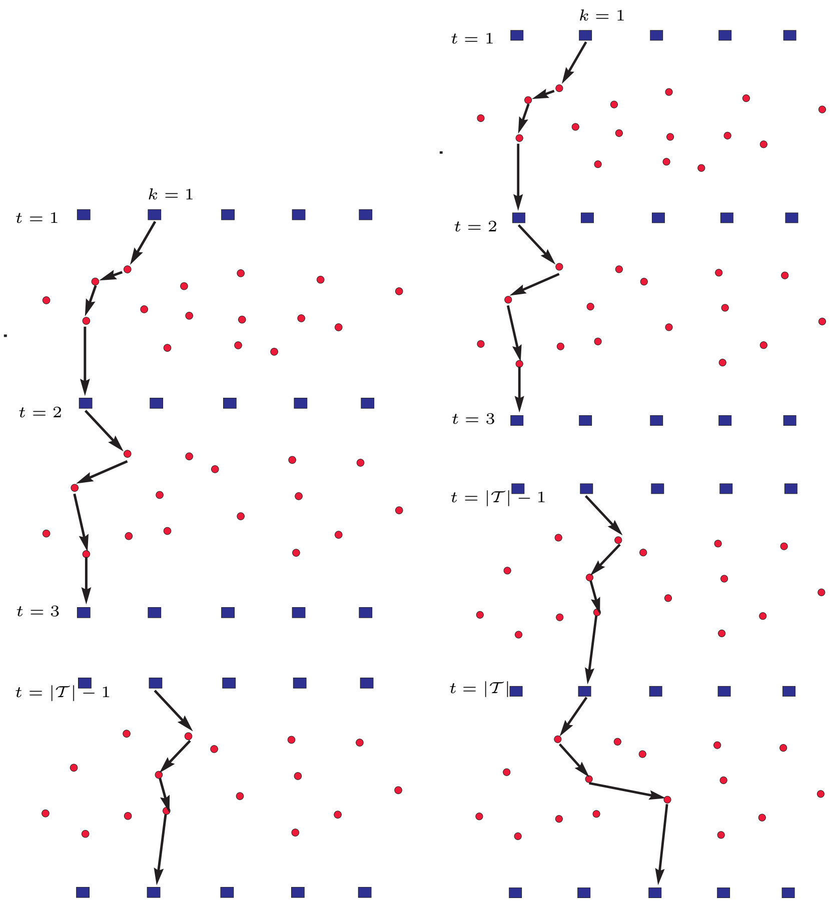
If a TS other than the closest TS to both end points is chosen, the objective is no better. Thus, by proof of contradiction, we pick the closest TS to both end points for $t = 2$.

At the end of $t = 2$, $n$ is the last covered HS on the route when $t = 2$, and the stop time is $f_{n1}^{2}$. Observation 1 picks the closest TS to $n$. Assume another TS' is picked. Then, the new finish time is $f_{n1}^{2'}$, and the rest of the route is the same.

$$\begin{aligned}&v(S) - v(S')\\&= f_{n1}^{2} - f_{n1}^{2'}\\&\geq 0, \text{due to } d_{n,TS}^{2} < d_{n,TS'}^{2}\end{aligned}$$

If a TS other than the closest TS is chosen, the objective is no better. Thus, by proof of contradiction, we pick the closest TS for the end of $t = 2$.

**Step two**: assume Observation 1 is true when there are $|\mathcal{T}| - 1$ periods.




(a) Step two (b) Step three

Figure 10: Steps two and three.

**Step three**: based on **Step two**, we add one more period between $|T| - m - 1$ and $|T| - m$ and get a total of $|\mathcal{T}|$ periods. Prove Observation 1 is true.

While the proof is very similar to that of Step one. To avoid redundancy, the detail is omitted.

Therefore, Observation 1 is proven true for one car.

Extending Observation 1 from one car to multiple cars is trivial. Since state trooper cars do not have overlapping coverage of HSs, each car is independent of each other. Observation 1 for multiple cars can be easily proven by induction. Hence, the proof is complete. ■

# 3: A MULTI-PERIOD HOME CARE SCHEDULING PROBLEM WITH WORK BALANCE

## 3.1 INTRODUCTION

In home care, caregivers travel to the homes of patients who request medical services due to acute illnesses, long-term health complications, permanent disabilities, or terminal illnesses. The use of home care service has seen a steady upward trend over the past decade. In 2009, annual expenditures for home health care were estimated to be \$72.2 billion. In 2010, approximately 12 million individuals received care from more than $33,000$ providers (National Association for Home Care, 2010). The growing trend of home care can be explained by the following reasons. First, fertility rates are lower than death rates, and thus there are not enough young people to take care of the elderly people. Second, as life expectancy increases, more elderly people need care. Third, compared with hospitalization and nursing homes, home care is the most economical choice. National Association for Home Care (2010) reported that hospitals cost \$$6,200$ per person per day, nursing homes \$622 per person per day, and home care \$135 per visit in 2009.

In 2006 alone, as National Association for Home Care (2008) pointed out, the total distance all the home caregivers travelled to serve their patients was approximately 5 billion miles. If we are able to improve the efficiency even as marginal as 1%, the savings in miles driven would be 50 million. As the number of home care recipients increases, more savings may be achieved through the design of a more efficient home health care scheduling system. With escalating gasoline prices, the savings in traveling distance will eventually translate into huge economic savings for home care providers and a relief for the fragile health care system.

In this paper, we focus on the operations of home care agencies (HCAs). They satisfy the needs of their patients by assigning appropriate caregivers during certain prescribed times over a recovery horizon while keeping the routing and staffing costs in check. Scheduling home caregivers is not trivial due to interwoven health care related constraints. Apart from the classical vehicle routing constraints, the additional complexity stems from the unique service characteristics of home health care: qualification-need match, blood test, caregiver-patient continuity, workload balance, synchronized visit requests by one patient, and sequential visit requests by one patient in one day.

- **Qualification-need match**: patient needs may include, but not limited to: i) cooking and light housekeeping, ii) laundry and change of bed linens, iii) grocery shopping and errands, iv) companionship and range of motion exercise, v) bathing, dressing and grooming assistance, vi) transportation to doctor appointments, supermarket, pharmacy, vii) assisting with walking and transfer from bed to wheelchair, viii) status reporting to family, and ix) medication reminders. Caregivers are generally categorized into staff nurse, registered nurse, physical therapist, occupational therapist, registered dietitian, and home health aide. Assigned caregiver qualifications should match with patient needs.

- **Blood test**: for the diagnosis and treatment of certain diseases, blood test is usually necessary. However, blood samples only remain viable for 2 hours and sometimes they can only be analyzed in the specialized labs which may not be close to the patients' homes, which could significantly affect the design of routes and schedules of the caregivers.

- **Continuity**: patients usually prefer being treated by the same caregivers, and caregivers also prefer treating the same patients.

- **Workload balance**: HCAs want to increase employee satisfactions by balancing the caregivers' workloads.

- **Synchronization**: certain patients' needs require services from multiple caregivers simultaneously. For example, an overweight patient may need multiple caregivers to maneuver.

- **Precedence**: certain patients' needs require sequential services from caregivers of different specializations. For example, a patient may need medication before/after lunch. A staff nurse gives him medication, while a home health aide cooks and feeds him lunch. Therefore, they cannot work simultaneously, and one should finish before the other one starts while the order does not really matter in this example. As a matter of fact, precedence constraints include the synchronization constraints, but we refer precedence to other situations in this paper due to the uniqueness of synchronization.

The research questions are i) how patients should be assigned to caregivers and ii) how caregivers should route so that the total traveling cost is minimized and the service level is guaranteed.

The literature terms the problem which combines assigning caregivers to patients and scheduling visits to patients' homes as the home care scheduling problem (HCSP). Our problem has workload balance and continuity constraints which link multiple periods together, and thus we cannot decompose our problem by time period as some previous studies did (Bredström and Rönnqvist, 2008; Eveborn et al., 2006; Rasmussen et al., 2012). We propose a novel model and name it the home care scheduling problem with workload balance (HCSP/WB). The HCSP/WB shares some similarities with the periodic vehicle routing problem (PVRP). For example, both the HCSP/WB and PVRP study the routing and scheduling of caregivers/vehicles to visit the patients/customers while satisfying the work hour limits and the visit frequency constraints. Also, the objectives in both problems are to minimize the total traveling distance and balance the workload over the scheduling horizon. However, our problem is more complicated than PVRP and is rarely studied in the previous literature. The major differences are: i) the service-specific constraints for our problem are not common in PVRP; and ii) our problem has a more flexible combination of visiting schemes. For instance, with a visit frequency of twice a week, PVRP may only allow for two combinations of delivery dates: (Monday, Wednesday) and (Tuesday, Thursday); in contrast, we can choose any two combinations out of seven, $\binom{7}{2} = 21$. As the frequency increases from twice a week to twice a month, the difference is substantial with regards to the number of options.

We apply column generation to solve the proposed model. Since HCAs admit and discharge patients on a daily basis, we solve our problem in a rolling horizon scheme. We execute the visiting plan according to the first day of the horizon, then move forward one day by updating admissions, discharges, and visits satisfied or not, and solve for the next horizon in the same way.

The remainder of this paper is structured as follows. In Section 3.2, we conduct a literature review on the HCSP and HCSP with workload balance. Section 3.3 presents our MILP model with several assumptions. We explain the column generation algorithm with details in Section 3.4. The numerical results and several managerial insights are included in Section 3.5. Finally, Section 3.6 concludes the paper and points out several future research directions.

## 3.2 LITERATURE REVIEW

The HCSP refers to the scheduling of caregivers who provide health services in patients' homes. The HCSP is becoming more and more important with the increased aging of the society. However, only limited research has been done using OR techniques to solve the HCSP. There are mainly two reasons: 1) home health care is an emerging area and has just started attracting OR researchers' attentions in recent years; and 2) the HCSP combines two well-known NP-hard problems – scheduling problem and vehicle routing problem, which makes the HCSP extremely hard to solve. In this section, we review several HCSPs in the OR literature with a clear presentation of their modeling similarities and differences and specifically focus on the HCSP/WB literature which provides direct inspiration to our problem.

Eveborn et al. (2006) is the first study which incorporates synchronization constraints in their HCSP model. Synchronization is unique in home health care settings in which two caregivers are required to visit some patients at the same time. Bredström and Rönnqvist (2008) extend the previous work to include temporal precedence constraints. For instance, medication has to be given by a caregiver before or after food. Their objective is to minimize a weighted sum of a continuity measure, the traveling cost, and a work balance measure. They use optimization-based heuristic to solve the problem, and the heuristic restricts the branch-and-bound tree to smaller size, yet big enough to improve the best known feasible solution. They show that separating synchronization constraints from the general precedence constraints and including them explicitly in the model help reduce the problem size, thus significantly improving the solution quality. More recently, Rasmussen et al. (2012) generalize both synchronization and temporal precedence constraints under five temporal dependency situations. They decompose the HCSP using Dantzig-Wolfe method and model it as a set partitioning problem with side constraints. Then they solve the model using dynamic column generation in a branch-and-price framework. Our problem differs from the aforementioned works by incorporating more realistic constraints and considering multi-period scheduling scheme, thus reflecting more complexities in the modeling perspective.

To the best of our knowledge, only Begur et al. (1997) and Nickel et al. (2012) study the HCSP/WB literature before. Begur et al. (1997) model the HCSP/WB as an MILP. However, since they argue that the agency under their study feels that their model is sufficient to capture the essence of the relevant issues involved, they restrict the visit date pattern in the same way as PVRP, and come up with an ad hoc model, explicitly specifying the allowable number of visits each week to once, twice, or three times. They develop a decision support system which can provide routes for caregivers by using GIS and embedded heuristic solver. Nickel et al. (2012) seek a weekly optimal plan, and therefore they produce the master schedule first and then adjust the operational daily plan to incorporate last minute changes. They do not have a mathematical model, and they factor in the continuity constraint by attaching it with some arbitrary weight in the objective and solve it heuristically in two stages. In the first stage, a constraint programming heuristic guarantees the quick calculation of a feasible solution. Afterwards, given the allowed total runtime, if further computation time is available, then an adaptive large neighborhood search (LNS) seeks to improve the initial solution.

In addition, we render it necessary to review the rolling horizon technique which we utilize in this paper. It has been successfully demonstrated by Bostel et al. (2008). They consider planning and routing technician visits to customers in the field, for maintenance or service logistics activities undertaken by utilities. The plans must be designed over a multi-period, rolling horizon and updated daily. They develop a memetic algorithm and a column generation/branch and bound heuristic in order to optimize an initial plan over a static horizon. They then adapt the procedures to cope with a rolling horizon, when a new plan has to be determined after the execution of each first daily period of the previous plan. Even though they do not have any home care specific constraints due to the different context, how they approach the multi-period facet of their problem is applicable to us.

## 3.3 GENERAL MODEL

The current scheduling practice at most HCAs is ad hoc-based and not efficient. Specifically, in the current practice, patients are manually assigned to caregivers by an experienced nurse, with the purpose of matching patient needs with caregiver qualifications. After caregivers are assigned to patients with different needs, they need to make their own schedules in an ad hoc way with considerations of the requested visit frequencies and time constraints. The whole scheduling process is currently done based on personal experience and even personal preference, thus inevitably causing a waste of resources and even poor care quality. In addition, the aforementioned service-related constraints further complicate the manual scheduling process.

To eliminate the waste, we take the approach of mathematical modeling.

### 3.3.1 Problem Definition

In October 2000, the Centers for Medicare and Medicaid Services (CMS) adopted a prospective payment system, which pays HCAs a predetermined rate for each 60-day episode of home health care. Consequently, most HCAs, including the one under our study, admit home care patients for a fixed length of 60 days. If a patient needs care for longer, his treatment can be extended for another episode. Meanwhile, our HCA checks its customer satisfaction and employee satisfaction every week. Considering both aspects, we can set our planning horizon $|\mathcal{P}|$ to either 60 or 7 days; since shorter period length results in a smaller problem size, $|\mathcal{P}|$ is set to 7. Each day is treated as one period, indexed by $p \in \mathcal{P}$.

In our study, the HCA maintains a list of active patients. Since some patients may send out synchronized requests or sequential requests, we define a service as one type of care to one patient performed by one caregiver. We use $\mathcal{J}$ to denote the set of service requests, and use $j$ to denote one service request, $j \in \mathcal{J}$. Each service is categorized by the patient name, his address, required caregiver type $q_j$, service time $s_j$, and visit frequencies $f_{jo}$. $f_{jo}$ has two indices, one for service and the other for numbering. The second index is needed because visit frequency may vary from day to day, week to week, and month to month. $o$ starts numbering from 1 till the total counts of visits for a request, $o \in \{1, 2, \ldots\}$, which is denoted as set $\mathcal{P}_j$; each visit is given a time slot $(p_{sjo},\ p_{ejo})$. For example, for a surgery recovery, a service request may require daily visit in the first week, and as the patient gradually recovers, two visits may be sufficient in the second week. This information can be expressed as $(p_{sjo},\ p_{ejo}) = (o,\ o),\ f_{jo} = 1,\ \forall o \in \{1, 2, 3, 4, 5, 6, 7\}$, and $(p_{sj8},\ p_{ej8}) = (8,\ 14),\ f_{j8} = 2$. One important feature about the visiting frequency is that a request is allowed to be visited at most once a day, due to the insurance requirement.

There are some services with special needs. Some services require blood tests, denoted as set $\mathcal{J}_\alpha \subset \mathcal{J}$. There are $|\mathcal{A}|$ labs, indexed by $a$. $D_a$ is the location for lab $a$. Some pairs of services $(i, j)$ by a patient demand multiple caregivers to be present simultaneously, denoted as $\mathcal{J}_\gamma$. Some other pairs of services $(i, j)$ have precedence relationship, denoted as

$\mathcal{J}_\delta$. Parameter $g_{ij}$ defines the time difference between $i$ and $j$. If $g_{ij} > 0$, $i$ ends $g_{ij}$ minutes before $j$ starts.

In addition to the patient information above, the HCA also keeps the caregiver information. $\mathcal{K}$ is the set of caregivers, and each caregiver is indexed by $k \in \mathcal{K}$. A caregiver has four attributes: her competence $e_{kj}$, the days when she is available to work $h_{pk}$, her shift length $H_{pk}$, and home address $D_k$. $e_{kj} = 1$ if caregiver $k$ has the necessary qualification to satisfy service $j$; 0 otherwise. $h_{pk} = 1$ if caregiver $k$ is available on day $p$; 0 otherwise. $D_k$ is where $k$ starts her work every day. $E_k$ is the dummy location for her to end the day, except for the first and the third days of each month, $\mathcal{P}_0$. On these days, a group meeting is held at the agency in the afternoon. $E_H$ is the location of the agency. As we mentioned before, this HCA wants to equilibrate workloads among its employees. The targeted workload $L$ is approximated by taking the ceiling function value of the ratio of the total number of visits during days $p \in \mathcal{P}$ over that of caregivers, that is, $\lceil \frac{\sum_{j\in\mathcal{J}} \sum_{\{o\in\mathcal{P}_j:1\le p\le|\mathcal{P}|\}} f_{jo}}{|\mathcal{K}|} \rceil$.

$\{\mathcal{J} \cup \bigcup_{k\in\mathcal{K}} (D_k \cup E_k) \cup E_H \cup \bigcup_{a\in\mathcal{A}} D_a\}$ forms the vertex set $\mathcal{V}$. Our underlying network is composed of these vertices connected. Travel time between all pairs of vertices is represented by parameter $t_{ij}$ (in minutes), $i,\ j \in \mathcal{V}$. We summarize all the sets, indexes, and parameters in Table 11.

In our HCSP model, we make the following assumptions:

1. lunch break is not considered;
2. one caregiver is categorized as one type;
3. the urban speed limit is 30 miles per hour, set as the caregivers' travel speed;
4. our HCA only approves caregivers of unexpected asking for a leave or calling in sick at least one day ahead of time; and
5. caregivers' workloads are independent of their qualifications.

Table 11: Sets, indexes, and parameters.

| | |
|---|---|
| *Sets*: | |
| $\mathcal{P}$ | period set. |
| $\mathcal{P}_0$ | period subset when caregivers have an agency-wide meeting. |
| $\mathcal{J}$ | service set. |
| $\mathcal{K}$ | caregiver set. |
| $\mathcal{A}$ | lab set. |
| $\mathcal{J}_\alpha$ | subset of services with blood tests. |
| $\mathcal{J}_\gamma$ | synchronization set. |
| $\mathcal{J}_\delta$ | precedence set. |
| $\mathcal{V}$ | vertex set. |
| *Indexes*: | |
| $p$ | period index, $p \in \mathcal{P}$. |
| $i, j$ | service index, $i, j \in \mathcal{J}$. |
| $k$ | caregiver index, $k \in \mathcal{K}$. |
| $a$ | lab index. |
| *Parameters*: | |
| $q_j$ | required caregiver type of service $j$. |
| $s_j$ | service time of service $j$. |
| $f_{jo}$ | visit frequency of the $o$th visit of service $j$, and each time slot ($p_{sjo}$, $p_{ejo}$). |
| $E_H$ | the agency location. |
| $D_a$ | lab location $a$. |
| $D_k$ | caregivers $k$'s home address where she starts her work. |
| $E_k$ | caregiver $k$'s dummy home address where she ends her work. |
| $g_{ij}$ | the precedence between services $i$ and $j$ in minutes. |
| $e_{kj}$ | 1 if caregiver $k$ is capable of serving $j$; 0 otherwise. |
| $h_{pk}$ | 1 if caregiver $k$ is available on day $p$; 0 otherwise. |
| $t_{ij}$ | travel time between services $i$ and $j$ in minutes. |
| $L$ | average workload. |

## Decision Variables

| | |
|---|---|
| $X_{pkij}$ | binary decision variable, $X_{pkij} = 1$ if caregiver $k \in \mathcal{K}$ travels from vertex $i \in \mathcal{V}$ to $j \in \mathcal{V}$, $i \neq j$ on day $p \in \mathcal{P}$. |
| $B_{pj}$ | continuous decision variable, indicating the starting time of each visit $j \in \mathcal{V}$ on day $p \in \mathcal{P}$. |
| $Y_{pij}$ | binary decision variable, $Y_{pij} = 1$ if request $i \in \mathcal{J}$ is performed before $j \in \mathcal{J}$ on day $p \in \mathcal{P}$. |
| $U_{kj}$ | binary decision variable, $U_{kj} = 1$ if request $j \in \mathcal{J}$ is assigned to caregiver $k \in \mathcal{K}$. |

### Objective

Minimize the total travel cost:

$$\text{Minimize} \sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}\setminus i} \mathcal{G} t_{ij} X_{pkij}.$$

where $\mathcal{G}$ represents the unit travel cost ($ per minute).

### Pre-process

Before introducing the actual constraints, we can pre-process $X_{pkij}$ by checking the match level of the competence of caregiver $k$ and request $j$, that is, $e_{kj}$ and her availability $h_{pk}$:

$$X_{pkij} \le U_{kj} \le e_{kj}, \qquad \forall p \in \mathcal{P},\ i \in \mathcal{V},\ j \in \mathcal{J},\ k \in \mathcal{K}. \tag{23}$$

$$X_{pkij} \le h_{pk}, \qquad \forall p \in \mathcal{P},\ i, j \in \mathcal{V},\ k \in \mathcal{K}. \tag{24}$$

### Route Structure Related Constraints

The first set of constraints is to build valid routes.

$$\sum_{j\in\mathcal{V}} X_{pkD_k,j} = h_{pk}, \qquad \forall p \in \mathcal{P},\ k \in \mathcal{K}. \tag{25}$$

$$\sum_{j\in\mathcal{V}\setminus i} X_{pkji} = \sum_{j\in\mathcal{V}\setminus i} X_{pkij} \le h_{pk}, \qquad \forall p \in \mathcal{P},\ i \in \mathcal{J} \cup \bigcup_{a\in\mathcal{A}} D_a,\ k \in \mathcal{K}. \tag{26}$$

$$\sum_{j\in\mathcal{V}} X_{pkj,E_k} = h_{pk}, \qquad \forall p \in \mathcal{P} \setminus \mathcal{P}_0,\ k \in \mathcal{K}. \tag{27}$$

$$\sum_{j\in\mathcal{V}} X_{pkj,E_H} = h_{pk}, \qquad \forall p \in \mathcal{P}_0,\ k \in \mathcal{K}. \tag{28}$$

Constraints (25)–(28) specify how caregivers travel. If one caregiver is available, she starts from her home at the beginning of a day, provides services, and comes back home at the end of the day, according to (25), (26), and (27), respectively; except on days $P_0$, all available caregivers end at the agency after their monthly meeting, according to (28). Note that Constraints (26) also play the essential role of ruling out the possibility that visits are lumped into one day.

**Request Satisfaction Constraints**

If a service is in the next few days, it has a detailed daily visit frequency; if it is far in the future, it has an aggregated weekly or monthly visit frequency instead.

$$\sum_{p=p_{sjo}}^{p_{ejo}} \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V} \setminus j} X_{pkij} = f_{jo}, \qquad \forall j \in \mathcal{J},\ o \in \mathcal{P}_j. \tag{29}$$

Constraints (29) can handle either daily or aggregate visit(s), ensuring each request must be served exactly at its desired frequencies. They are different by the span of the given time limits; daily/weekly/monthly visit(s) allow(s) for one day/week/month to be served.

**Schedule Feasibility Constraints**

$$B_{pj} - B_{pD_k} \leq H_{pk}, \qquad \forall p \in \mathcal{P},\ j \in E_H \cup E_k,\ k \in \mathcal{K}. \tag{30}$$

$$B_{pi} + s_i + t_{ij} \leq B_{pj} + M \times (1 - \sum_{k \in \mathcal{K}} X_{pkij}), \qquad \forall p \in \mathcal{P},\ i, j \in \mathcal{V}. \tag{31}$$

Constraints (30) enforce that no caregiver works longer than her shift length. Constraints (31) formulate that there is enough time for a caregiver to travel from one request to another, which is linearized by utilizing the big number $M$, set to $\max_{p \in \mathcal{P},\, k \in \mathcal{K}} \{H_{pk}\} = 600$ minutes. All the aforementioned constraints are standard routing constraints, while the following constraints are home care specific.

**Blood Sample Constraints**

$$\sum_{i \in \mathcal{J} \setminus j} X_{pkij} = \sum_{a \in \mathcal{A}} X_{pkj,D_a}, \qquad \forall p \in \mathcal{P},\ j \in \mathcal{J}_\alpha,\ k \in \mathcal{K}. \tag{32}$$

After blood samples are taken, the caregiver immediately takes them back to a lab, which is achieved by Constraints (32). Even though blood samples stay viable for 2 hours, the HCA does not allow caregivers to squeeze in any service before the trip to a lab in order to avoid contaminations.

**Workload Balance Constraints**

$$L - \epsilon \leq \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{J} \setminus i} X_{pkij} \leq L + \epsilon, \qquad \forall k \in \mathcal{K}. \tag{33}$$

Constraints (33) allow minor variations on the number of visits a caregiver makes in total. The variation limit is set to a small number, i.e., $\epsilon = 10$. Since we assume that caregivers' workloads are independent of their qualifications, the constraints do not depend on qualifications.

**Continuity Constraints**

$$\sum_{o \in \mathcal{P}_j} \sum_{p=p_{sjo}}^{p_{ejo}} \sum_{i \in \mathcal{J} \setminus j} X_{pkij} = \sum_{o \in \mathcal{P}_j} f_{jo} \times U_{kj}, \qquad \forall j \in \mathcal{J},\ k \in \mathcal{K}. \tag{34}$$

$$\sum_{k \in \mathcal{K}} U_{kj} = 1, \qquad \forall j \in \mathcal{J}. \tag{35}$$

Constraints (34)–(35) restrict that each requested home care service must be performed by the same caregiver during the entire time horizon of the treatment.

**Synchronization Constraints**

$$B_{pj} = B_{pi}, \qquad \forall p \in \mathcal{P},\ (i,\ j) \in \mathcal{J}_{\gamma}. \tag{36}$$

To link synchronized services, Constraints (36) guarantee multiple caregivers show up at the same time.

**Precedence Constraints**

In addition to services in synchronization, there are some services in precedence. When a patient is visited twice or more times in one day, we should make sure that precedence constraints are satisfied between those pairs of services involved.

To achieve this, we need another set of decision variables $Y_{pij}$. By definition, $Y_{pij}$ or $Y_{pji}$ can only be equal to 1 when both $\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{V}} X_{pkli} = 1$ and $\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{V}} X_{pklj} = 1$; or else, $Y_{pij} = Y_{pji} = 0$ for $i,\ j \in \mathcal{J}$. The following constraints enforce this relationship:

$$Y_{pij} + Y_{pji} \leq \sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{V}} X_{pkli}, \qquad \forall p \in \mathcal{P},\ (i,\ j) \in \mathcal{J}_{\delta},\ j > i. \tag{37}$$

$$Y_{pij} + Y_{pji} \leq \sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{V}} X_{pklj}, \qquad \forall p \in \mathcal{P},\ (i,\ j) \in \mathcal{J}_{\delta},\ j > i. \tag{38}$$

$$Y_{pij} + Y_{pji} \geq \sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{V}} X_{pkli} + \sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{V}} X_{pklj} - 1, \qquad \forall p \in \mathcal{P},\ (i,\ j) \in \mathcal{J}_{\delta},\ j > i. \tag{39}$$

We are only interested in those $(i, j)$ pairs corresponding to the same patient. Actually, Constraints (37)–(39) capture all pairs of services done within the same day, including synchronized services. Since Constraints (36) have ensured synchronization, the other constraints are:

$$B_{pi} + s_i + g_{ij} \le B_{pj} + M \times (1 - Y_{pij}), \qquad \forall p \in \mathcal{P},\ (i,\ j) \in \mathcal{J}_\delta,\ j > i, \tag{40}$$

$$B_{pj} + s_j + g_{ij} \le B_{pi} + M \times (1 - Y_{pji}), \qquad \forall p \in \mathcal{P},\ (j,\ i) \in \mathcal{J}_\delta,\ j > i. \tag{41}$$

If service $i$ is performed before $j$, then Constraints (40) ensure that the treatment of $j$ can only start after the treatment of $i$ has been finished for $g_{ij}$ minutes; or the other way around in the next set of constraints. In this paper, we set $g_{ij} = 0$ since the HCA under study has only encountered pairs of services without any overlap before, but these constraints are suitable for more generic situations.

**Integrality and Non-negativity Constraints**

Finally, we state continuous and binary variables:

$$X_{pkij}, Y_{pij}, U_{kj} \in \{0, 1\}, B_{pj} \ge 0, \qquad \forall p \in \mathcal{P},\ i,\ j \in \mathcal{V},\ k \in \mathcal{K}. \tag{42}$$

### 3.3.2 Overall Model

Minimizing the total travel cost, subject to Constraints (23)–(42), constitutes our overall model of the HCSP/WB, categorized as an MILP. The condensed formulation is provided in the appendix. If the visit frequency pattern is fixed as a constant and all home care specific constraints are discarded, then the HCSP/WB reduces to a multi-depot PVRP. PVRP is shown to be NP-hard (Gaudioso and Paletta, 1992), and thus our problem is NP-hard too. NP-hard problems are notorious for their computational intractability using exact solution algorithms. In this paper, we propose to solve our problem using a decomposition method.

### 3.3.3 Model Variant

We would like to introduce a model variant to solve a specific problem encountered in real practice. The proposed model is built upon the status quo of the HCA of interest, in which no caregiver is equipped with a centrifuge to perform blood testing on site. The HCA wonders whether to purchase centrifuges for its caregivers. A centrifuge costs $1000 per piece. If a caregiver owns it, she does not need to take blood samples to a lab any more, saving the traveling cost. The question is whether the total saving is sufficient to cover the investment. It can be easily answered by our model and its variant with only a minor modification. We eliminate Constraints (32) and solve the rest of the model, and the saving is equivalent to the difference between objective function values with and without Constraints (32). If the total saving outnumbers the total centrifuge cost for all caregivers, then it is beneficiary for the HCA to make this investment.

## 3.4 COLUMN GENERATION

Column generation is a way of solving a linear programming problem that adds columns during the pricing phase of the simplex method of solving the problem. Dantzig and Wolfe (1960) adapted it to linear programming problems with a decomposable structure, called Dantzig-Wolfe decomposition. Barnhart et al. (2007) summarized the methodology of branch-and-price, which applies column generation techniques in integer programming solution methods. By definition, column generation is the back engine of both Dantzig-Wolfe decomposition and branch-and-price. Column generation has been used extensively in vehicle routing and crew scheduling problems, whose superior results were summarized by Desrosiers et al. (1995). In the context of the HCSP, which in essence combines the vehicle routing problem and the scheduling problem, Rasmussen et al. (2012) used Dantzig-Wolfe decomposition and demonstrated its efficiency by showing good results on both real-life problem instances and on generated test instances inspired by realistic settings. The success in the utilization of column generation in the VRP and scheduling literature as well as the most relevant paper suggests the same approach for our problem.

In our LP-based column generation framework, the master problem is composed of the constraints which link multiple periods together, including request satisfaction, workload balance, and continuity while relaxing the integrality of the decision variables; other constraints remain in the subproblem. Due to the size of our problem, the master problem has a huge number of columns to be handled efficiently. Instead of explicitly adding all possible columns, we use the subproblem to identify columns to enter the basis. The process starts by first generating an initial feasible solution using a heuristic, which feeds the input to the master problem. The master problem passes on dual solutions to the objective function of the subproblem. The subproblem generates feasible home caregiver schedules, which are solvable period by period. If the objective function value of the subproblem is positive, the schedule of this period is the next column to enter the basis in the master problem, increasing its number of columns by one. Then, resolve the master problem with an enlarged feasible region. This iteration terminates when none of the subproblems has a positive objective function value, indicating optimality.

Next, we present the implementation details of the column generation algorithm.

#### 3.4.1 Decomposition

A feasible schedule $r$ on day $p \in \mathcal{P}$ is defined as a collection of routes for all caregivers, with each caregiver $k$ starting from her home $D_k$, visiting patients, and ending at her home $E_k$ or the agency $E_H$. The schedule includes the starting time of each visit as well. The parameter $c_{pr}$ is the cost of schedule $r$ on day $p$, $c_{pr} = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V} \setminus i} \mathcal{G} t_{ij} X_{pkij}$. The binary parameter $a_{pkrj} = 1$ if request $j$ is included in schedule $r$ of route $k$ on day $p$; 0 otherwise.

##### 3.4.1.1 Master Problem (MP)

$\mathcal{R}_p$ is the set of all feasible schedules on day $p$. In the master problem of our decomposition, we introduce the binary decision variable $\lambda_{pr}$. It is equal to 1 if schedule $r$ is used on day $p$; 0 otherwise. Based on the definition, we have the following relationship: $\sum_{i \in \mathcal{V}/j} X_{pkij} =$

$\sum_{r \in \mathcal{R}_p} a_{pkrj} \times \lambda_{pr}$. Utilizing this relationship, we formulate our MP as follows:

$$\text{Minimize} \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}_p} c_{pr} \times \lambda_{pr}.$$

$$\sum_{p=p_{sjo}}^{p_{ejo}} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_p} a_{pkrj} \times \lambda_{pr} = f_{jo}, \qquad \forall j \in \mathcal{J},\ o \in \mathcal{P}_j. \tag{43}$$

$$L - \epsilon \leq \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}_p} \sum_{j \in \mathcal{J}} a_{pkrj} \times \lambda_{pr} \leq L + \epsilon, \qquad \forall k \in \mathcal{K}. \tag{44}$$

$$\sum_{o \in \mathcal{P}_j} \sum_{p=p_{sjo}}^{p_{ejo}} \sum_{r \in \mathcal{R}_p} a_{pkrj} \times \lambda_{pr} = \sum_{o \in \mathcal{P}_j} f_{jo} \times U_{kj}, \qquad \forall j \in \mathcal{J},\ k \in \mathcal{K}. \tag{45}$$

$$\sum_{k \in \mathcal{K}} U_{kj} = 1, \qquad \forall j \in \mathcal{J}. \tag{46}$$

$$\sum_{r \in \mathcal{R}_p} \lambda_{pr} = 1, \qquad \forall p \in \mathcal{P}. \tag{47}$$

$$\lambda_{pr}, U_{kj} \in \{0, 1\}, \qquad \forall p \in \mathcal{P},\ j \in \mathcal{J},\ k \in \mathcal{K},\ r \in \mathcal{R}_p. \tag{48}$$

If the size of set $\mathcal{R}_p$ is small, the MP can be solved via commercial software. However, this is not our case, and thus we restrict $\mathcal{R}_p$ to only contain a small subset of the promising schedules, denoted by $\mathcal{R}'_p$.

#### 3.4.1.2 Relaxed and Restricted Master Problem (RRMP)

In addition to this restriction, since column generation solves LP problems, we relax the integrality of $\lambda_{pr}$. This series of transformations complete the relaxation and restriction process, and the formulation of our RRMP is provided. For the ease of obtaining dual solutions for the subproblem, we split Constraints (44) into separate lines.

$$\text{Minimize} \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}'_p} c_{pr} \times \lambda_{pr}.$$

$$\sum_{p=p_{sjo}}^{p_{ejo}} \sum_{k\in\mathcal{K}} \sum_{r\in\mathcal{R}'_p} a_{pkrj} \times \lambda_{pr} = f_{jo}, \qquad \forall j \in \mathcal{J},\ o \in \mathcal{P}_j. \tag{49}$$

$$L - \epsilon \leq \sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}'_p} \sum_{j\in\mathcal{J}} a_{pkrj} \times \lambda_{pr}, \qquad \forall k \in \mathcal{K}. \tag{50}$$

$$\sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}'_p} \sum_{j\in\mathcal{J}} a_{pkrj} \times \lambda_{pr} \leq L + \epsilon, \qquad \forall k \in \mathcal{K}. \tag{51}$$

$$\sum_{o\in\mathcal{P}_j} \sum_{p=p_{sjo}}^{p_{ejo}} \sum_{r\in\mathcal{R}'_p} a_{pkrj} \times \lambda_{pr} = \sum_{o\in\mathcal{P}_j} f_{jo} \times U_{kj}, \qquad \forall j \in \mathcal{J},\ k \in \mathcal{K}. \tag{52}$$

$$\sum_{k\in\mathcal{K}} U_{kj} = 1, \qquad \forall j \in \mathcal{J}. \tag{53}$$

$$\sum_{r\in\mathcal{R}'_p} \lambda_{pr} = 1, \qquad \forall p \in \mathcal{P}. \tag{54}$$

$$0 \leq \lambda_{pr}, U_{kj} \leq 1, \qquad \forall j \in \mathcal{J},\ k \in \mathcal{K},\ r \in \mathcal{R}'_p. \tag{55}$$

For each primal solution to the RRMP, we obtain a dual solution $[\pi, \zeta, \nu, \eta, \omega]$, where $\pi_{jo}$, $\zeta_k$, $\nu_k$, $\eta_{jk}$, and $\omega_p$ are the dual variables for Constraints (49) – (52) and (54), respectively. Dual solutions play their roles through the reduced cost, denoted by $\hat{c}_{pr}$:

$$\begin{aligned}
&\hat{c}_{pr} \\
&= \sum_{k\in\mathcal{K}} [ \sum_{\{j\in\mathcal{J}, o\in\mathcal{P}_j : p_{sjo} \leq p \leq p_{ejo}\}} (\pi_{jo} + \eta_{jk}) a_{pkrj} + \zeta_k (- \sum_{j\in\mathcal{J}} a_{pkrj}) + \nu_k \sum_{j\in\mathcal{J}} a_{pkrj} ] + \omega_p - c_{pr} \\
&= \sum_{k\in\mathcal{K}} [ \sum_{\{j\in\mathcal{J}, o\in\mathcal{P}_j : p_{sjo} \leq p \leq p_{ejo}\}} (\pi_{jo} + \eta_{jk}) a_{pkrj} + \zeta_k (- \sum_{j\in\mathcal{J}} a_{pkrj}) + \nu_k \sum_{j\in\mathcal{J}} a_{pkrj} ] + \omega_p - c_{pr} \\
&= \sum_{k\in\mathcal{K}} [ \sum_{\{j\in\mathcal{J}, o\in\mathcal{P}_j : p_{sjo} \leq p \leq p_{ejo}\}} (\pi_{jo} + \eta_{jk}) \sum_{i\in\mathcal{V}\setminus j} X_{pkij} - \zeta_k \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{V}\setminus j} X_{pkij} + \nu_k \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{V}\setminus j} X_{pkij} - \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{V}\setminus i} \mathcal{G} t_{ij} X_{pkij} ] + \omega \\
&= \sum_{k\in\mathcal{K}} [ \sum_{\{j\in\mathcal{J}, o\in\mathcal{P}_j : p_{sjo} \leq p \leq p_{ejo}\}} (\pi_{jo} + \eta_{jk}) \sum_{i\in\mathcal{V}\setminus j} X_{pkij} - (\zeta_k - \nu_k) \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{V}\setminus j} X_{pkij} - \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{V}\setminus i} \mathcal{G} t_{ij} X_{pkij} ] + \omega_p,
\end{aligned}$$

due to $a_{pkrj} = \sum_{i\in\mathcal{V}\setminus j} X_{pkij}, \forall p \in \mathcal{P}, j \in \mathcal{J}, k \in \mathcal{K}, r = |\mathcal{R}'_p| + 1$.

3.4.1.3 Subproblem

The rest of constraints, which are left out of the master problem, go to the subproblem. It turns out that the subproblem is decomposable period by period. For the ease of presentation, we drop $p$ index in the decision variables, and thus we have $X_{kij}$ and $B_j$. In addition, the size of each subproblem can be reduced further if we divide each day's service set to requests which are allowed to be visited on this day. We define the service set on day $p$ $\tilde{\mathcal{J}}^p$, and define the network as $\tilde{\mathcal{V}}^p = \{\tilde{\mathcal{J}}^p \cup \bigcup_{k\in\mathcal{K}}(D_k \cup E_k) \cup E_H \cup \bigcup_{a\in\mathcal{A}} D_a\}$.

$$\text{Maximize } \sum_{k\in\mathcal{K}}[\sum_{\{j\in\tilde{\mathcal{J}}^p, o\in\mathcal{P}_j : p_{sjo}\leq p\leq p_{ejo}\}}(\pi_{jo}+\eta_{jk})\sum_{i\in\tilde{\mathcal{V}}^p\backslash j} X_{kij} - (\zeta_k - \nu_k)\sum_{j\in\tilde{\mathcal{J}}^p}\sum_{i\in\tilde{\mathcal{V}}^p\backslash j} X_{ij} - \sum_{i\in\tilde{\mathcal{V}}^p}\sum_{j\in\tilde{\mathcal{V}}^p\backslash i}\mathcal{G}t_{ij}X_{kij}].$$

$$X_{kij} \leq e_j, \qquad \forall i,j \in \tilde{\mathcal{J}}^p,\ k \in \mathcal{K}. \tag{56}$$

$$\sum_{j \in \tilde{\mathcal{V}}^p} X_{kD_k,j} = 1, \qquad \forall k \in \mathcal{K}. \tag{57}$$

$$\sum_{j \in \tilde{\mathcal{V}}^p \setminus i} X_{kji} = \sum_{j \in \tilde{\mathcal{V}}^p \setminus i} X_{kij} \leq 1, \qquad \forall i \in \tilde{\mathcal{J}}^p \cup \bigcup_{a \in \mathcal{A}} D_a,\ k \in \mathcal{K}. \tag{58}$$

$$\sum_{j \in \tilde{\mathcal{V}}^p} X_{kj,E_k} = 1, \qquad \forall k \in \mathcal{K}. \tag{59}$$

$$\sum_{j \in \tilde{\mathcal{V}}^p} X_{kj,E_H} = 1, \qquad \forall k \in \mathcal{K}. \tag{60}$$

$$B_j - B_{D_k} \leq H, \qquad \forall j \in E_H \cup E_k,\ k \in \mathcal{K}. \tag{61}$$

$$B_i + s_i + t_{ij} \leq B_j + M \times (1 - \sum_{k \in \mathcal{K}} X_{kij}), \qquad \forall i,j \in \tilde{\mathcal{V}}^p. \tag{62}$$

$$\sum_{i \in \tilde{\mathcal{J}}^p \setminus j} X_{kij} = \sum_{a \in \mathcal{A}} X_{kj,D_a}, \qquad \forall j \in \tilde{\mathcal{J}}^p_\alpha,\ k \in \mathcal{K}. \tag{63}$$

$$B_j = B_i, \qquad \forall (i,\ j) \in \mathcal{J}^p_\gamma. \tag{64}$$

$$Y_{ij} + Y_{ji} \leq \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{V}^p} X_{kli}, \qquad \forall (i,\ j) \in \mathcal{J}^p_\delta,\ j > i. \tag{65}$$

$$Y_{ij} + Y_{ji} \leq \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{V}^p} X_{klj}, \qquad \forall (i,\ j) \in \mathcal{J}^p_\delta,\ j > i. \tag{66}$$

$$Y_{ij} + Y_{ji} \geq \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{V}^p} X_{kli} + \sum_{k \in \mathcal{K}} \sum_{v \in \mathcal{V}^p} X_{klj} - 1, \qquad \forall (i,\ j) \in \mathcal{J}^p_\delta,\ j > i. \tag{67}$$

$$B_i + s_i + g_{ij} \leq B_j + M \times (1 - Y_{ij}), \qquad \forall (i,\ j) \in \mathcal{J}^p_\delta,\ j > i. \tag{68}$$

$$B_j + s_j + g_{ij} \leq B_i + M \times (1 - Y_{ji}), \qquad \forall (i,\ j) \in \mathcal{J}^p_\delta,\ j > i. \tag{69}$$

$$X_{kij} \in \{0,1\}, Y_{ij} \in \{0,1\}, B_j \geq 0, \qquad \forall i,\ j \in \tilde{\mathcal{V}}^p,\ k \in \mathcal{K}. \tag{70}$$

Our subproblem is basically a special case of the single period HCSP studied by Rasmussen et al. (2012) when every customer must be visited. If we solve the subproblem as suggested by Rasmussen et al. (2012), our runtime could be much longer than theirs with the same problem size. Since having an efficient solution to the subproblem of the column generation is critical to its successful implementation, we propose a faster heuristic to solve the HCSP/WB. We present the details of our algorithm in Section 1.0.1 .

Next, we discuss some other important procedures of our column generation implementation. Each time when a subproblem is solved for day $p$, its solution expands parameters $c_{pr}$ and $a_{pkrj}$ , $\forall j \in \mathcal{J},\ k \in \mathcal{K}$. When the subproblems are solved for all days, we obtain the information of $c_{pr}$, $\forall p \in \mathcal{P}$. Then, we can evaluate $\max_{p\in\mathcal{P}}\{\hat{c}_{pr}\}$ to check the optimality condition of the LP-based column generation. If its value is positive, any schedule with a positive subproblem objective enters the basis in the RRMP as a new column; or else, the RRMP is optimal. Due to the tailing effect of the column generation, we terminate the process prematurely when $\max_{p\in\mathcal{P}}\{\hat{c}_{pr}\}$ is small enough, i.e., 5. However, the solution to the RRMP is not necessarily integer. One final step of the column generation is to guarantee integer solutions. In the literature of the column generation, there are three options: simple rounding, getting integer solutions using heuristics, and the branch and price. Compared with the search-intensive second option and the computationally expensive third option, we choose simple rounding because of its simplicity and efficiency.

To summarize our solution approach, we offer a flowchart in Figure 11.

### 3.4.2 Initialization

As shown in Figure 11, the column generation starts from some feasible schedules, required by the input $\mathcal{R}'_p$. It should contain at least one feasible schedule for all $p$. To obtain a feasible schedule, we use a simple heuristic, which at the beginning generates schedules by considering all but the precedence constraints, then performs neighborhood searches to bring down the total travel cost, and at last updates the built schedules by incorporating the precedence constraints.

The schedules are constructed in five phases in sequence. Phase 1 adds the starting and stopping locations for each caregiver on those days when she is on duty. Phase 2 takes care of those requests with synchronized visits. First, we find a pair of capable and available caregivers by checking whether their skills match the requests and whether they are on

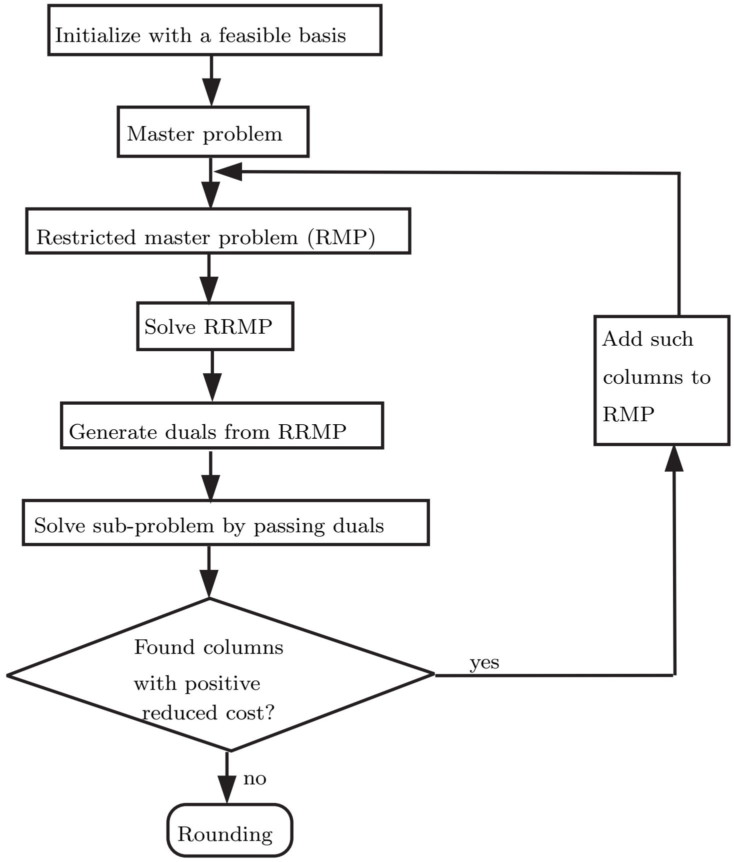



Figure 11: Flow chart for the column generation with the rounding procedure.

duty frequent enough to satisfy the frequency requirements. The check is done in the order of caregivers with the increasing number of assigned visits so that the workload can be balanced. Assign one synchronized request to one caregiver and assign the other one to the other caregiver so as to maintain continuity. Second, we find days and time when each visit is made in the entire horizon. The days are picked as the assigned two caregivers' earliest available working days within the allowed time slot while respecting the work hour limit. We add synchronized requests as the assigned caregivers' next visits on those picked days. On each picked day, the starting time is set the same, equal to the later arrival time of the two caregivers. Next, Phase 3 finds a competent caregiver to satisfy each unsatisfied request and adds each request at the end of the picked caregiver's route. Similar to the previous phase, we pick the dates and time by taking into considerations of the qualification-need match, availability, workload balance, continuity, and work hour limit. Phase 3 differs from the previous one in the facts that one caregiver is picked for only one request and the starting time of each visit is the arrival time of the picked caregiver.

Since the first three phases sequentially select a caregiver for a request, the date of each visit, and its arrival time, the schedules built are dependent on the order of the selections. In Phase 4, to improve the schedules, we change the selections of the caregiver, the date, or the sequence of each route, which determine the arrival time. Due to the workload and continuity constraints, our search space is restricted in the sense that reselecting a caregiver for a request reassigns it to a different caregiver for the entire horizon, which is very hard to achieve; whereas, reselecting days or reshuffling route sequence is relatively easier. Following this reselection guideline, we propose four neighborhood search algorithms:

**(i)** exchange two visits of a caregiver on each day,

**(ii)** exchange two visits of a caregiver on different days, on the condition that both days fall into the corresponding time slots of both visits,

**(iii)** remove one visit of a caregiver and find another place on her route on each day, and

**(iv)** remove one visit of a caregiver on one day and place it on her route on another day, on the condition that the newly picked date is also within the allowed time span.

If any of these algorithms leads to a lower travel cost, it is accepted. In other words, our acceptance rule is the first improvement. These algorithms are inspired by two of the most popular VRP operators, the exchange operator which swaps two locations on two routes and the relocate operator which removes one location from one route and places it into another route. For more details, we refer to the survey paper by Bräysy and Gendreau (2005).

Till now, we have not considered the precedence constraints, so the schedules built may have some patient(s) with overlapping visits which cannot be done by different caregivers at the same time. Therefore, the last phase is needed to fix any violation of the precedence constraints. Since violation only involves daily schedules, the most obvious way to fix it is to work on the overlapping visits at the same patient and separate their visit time apart. We accomplish this in Phase 5 by calling Algorithms 7 and 8. The pseudocode of the final phase is provided in Algorithm 6.

**Algorithm 6** Algorithm 5: Precedence

1: *Loop* $\leftarrow 0$;
2: **repeat**
3: *Loop* $= 0$;
4: **for** $p$ in $1 \dots |\mathcal{P}|$ **do**
5: **for** $k$ in $1 \dots |\mathcal{K}| - 1$ **do**
6: **for** $g$ in $k + 1 \dots |\mathcal{K}|$ **do**
7: **if** $h_{pk} = 1$, $h_{pg} = 1$ **then**
8: **for** $i$ in $2 \dots Rou_{pk}.size() - 2$ **do**
9: **for** $j$ in $2 \dots Rou_{pg}.size() - 2$ **do**
10: **if** $(i, j)$ in $\mathcal{J}_\delta$ **then**
11: **if** $i$ and $j$ overlaps in service time **then**
12: *Adjust* $\leftarrow 0$; *Dif* $\leftarrow$ the overlap time;
13: **if** $i$ starts after $j$ **then**
14: **Shift**$(p, k, i, Rou, Adjust, Dif, Loop)$;
15: **if** *Adjust* $= 0$ **then**
16: **Shift**$(p, g, j, Rou, Adjust,\ s_{Rou_{pki}} + s_{Rou_{pgj}} - Dif,\ Loop)$;
17: **end if**
18: **if** *Adjust* $= 0$ **then**
19: **Swap**$(p, k, i, B_{pi}, Rou, Adjust, Dif, Loop, g, j)$;
20: **end if**
21: **if** *Adjust* $= 0$ **then**
22: **Swap**$(p, g, j, B_{pj}, Rou, Adjust,\ s_{Rou_{pki}} + s_{Rou_{pgj}} - Dif,\ Loop, k, i)$;
23: **end if**
24: **else**
25: Exchange $k$ with $g$, and $i$ with $j$ and repeat steps 14-24;
26: **end if**
27: **end if**
28: **end if**
29: **end for**
30: **end for**
31: **end if**
32: **end for**
33: **end for**
34: **end for**
35: **until** *Loop* $= 1$

Algorithms 7 performs a neighborhood search to separate overlapping time of requests with the precedence constraints. In Phases 1-4, if a visit has synchronization requirement, we set the beginning time to the later arrival time; if a visit does not, we set the beginning time to the arrival time. The visits are added back to back without any breaks. We observe that if there is any idle time, it can only exist right before a synchronized visit or the last visit of each route. The existence of idle time leaves room for the beginning time of either violated visit to be shifted forward. If either time is shifted forward far enough such that the requests with precedence constraints do not overlap any more, then the violation is fixed to feasibility. The pseudocode is provided in Algorithm 7.

**Algorithm 7** Algorithm: Shift($p, k, i, Rou, Adjust, Dif, Loop$)

$w = i$;
**for** $w$; $w < Rou_{pk}.size()$; $w++$ **do**
  **if** $Rou_{pkw} \in \mathcal{J}_\alpha$ **then**
    Break;
  **end if**
**end for**
$Sum \leftarrow 0$;
**for** $z = i$; $z < w$; $z++$ **do**
  $Sum += t_{Rou_{pkz} Rou_{pkz+1}} + s_{Rou_{pkz}}$;
  $\Delta \leftarrow B_{pRou_{pkz+1}} - Sum$;
  **if** $Dif \leq \Delta$ **then**
    The begin time of $Rou_{pki}$ adds $Dif$;
    **for** $y = i + 1$; $y \leq z$; $y++$ **do**
      Adjust the begin time of $Rou_{pky}$;
    **end for**
    $Adjust = 1$; $Loop = 1$; Break;
  **end if**
**end for**

It is also possible that the work hour limit leaves no idle time in the violated routes. In this case, we propose Algorithm 8 to fix the violation. It swaps one of the violated visits with another non-synchronized visit on the same route so that the swapped visit's service time is changed to another time. Synchronized visit is not allowed to change due to the synchronization constraints. Swap would not violate any of the continuity, workload balance, or request frequency requirement. If the swap leads to the violated pair of visits without any overlap, the swap is accepted. The pseudocode is provided in Algorithm 8.

**Algorithm 8** Algorithm: Swap$(p, k, i, B_{pi}, Rou, Adjust, Dif, Loop, g, j)$

**if** $Rou_{pki}$ is not in $\mathcal{J}_\alpha$ **then**
  **for** $z = Rou_{pk}.size() - 2;\ z > 1;\ z - -$ **do**
    **if** $z \in \mathcal{J} \setminus \mathcal{J}_\alpha \setminus \mathcal{J}_\delta$ **then**
      **if** New begin time at $Rou_{pkz+1}$ and $Rou_{pki+1}$ are $\leq$ old time **then**
        **if** ($z > i$ and new begin time at $i$ − its old time $\geq Dif$) or ($z < i$ and old begin time at $i$ − its new time $\geq s_{Rou_{pki}} + s_{Rou_{pgj}} - Dif$) **then**
          Swap $Rou_{pki}$ and $Rou_{pkz}$;
          Update begin time at $Rou_{pkz}$, $Rou_{pkz+1}$, $Rou_{pki}$ and $Rou_{pki+1}$;
          $c_{k1} += \mathcal{G} \times (t_{Rou_{pkz-1}Rou_{pki}} + t_{Rou_{pki}Rou_{pkz+1}} + t_{Rou_{pki-1}Rou_{pkz}} + t_{Rou_{pkz}Rou_{pki+1}})$;
          $c_{k1} -= \mathcal{G} \times (t_{Rou_{pkz-1}Rou_{pkz}} + t_{Rou_{pkz}Rou_{pkz+1}} + t_{Rou_{pki-1}Rou_{pki}} + t_{Rou_{pki}Rou_{pki+1}})$;
          $Ajust = 1$; $Loop = 1$; Break;
        **end if**
      **end if**
    **end if**
  **end for**
**end if**

These fives phases complete our initialization step, outputting a feasible solution. This satisfies the requirement of a non-empty set $\mathcal{R}'_p$. Then, the RRMP is solved by the simplex method, whereas the subproblems need a heuristic solution approach.

### 3.4.3 Subproblem Heuristic

Since each subproblem needs to build up feasible routes while satisfying the work hour limit, synchronization, and precedence constraints, which resembles the initialization step, the previous subsection sheds light on how our subproblem is solved. As the initialization step, the subproblem is also solved in five phases. Phases 1 and 5 are exactly the same, while phases 2, 3, and 4 have some minor modifications because the constraints are not as stringent as before.

In the subproblem, there are no continuity or workload balance constraints, so the modification is to eliminate their corresponding restrictions on the selections of caregivers for requests in Phases 2 and 3. The rest of the algorithm remains the same. Phase 4 is also composed of four neighborhood search algorithms as in the previous subsection. Algorithms (i) and (iii) still apply to the subproblem, while Algorithms (ii) and (iv) do not apply any more since they involve multiple days. The continuity constraints are not present in the subproblem, and thus we can assign any request to any caregiver as long as the skill is matched. Therefore, we replace Algorithms (ii) and (iv) with two new neighborhood searches by allowing changes on different caregivers. These two algorithms are updated to:

**(ii)** exchange two visits of two caregivers on one day, and

**(iv)** remove one visit of a caregiver and place it on another caregiver's route on one day.

The acceptance rule stays the same, that is, the first improvement.

At last, our column generation ends with a simple rounding to obtain integer solutions. We propose to solve the HCSP/WB in a rolling horizon scheme, which is the final algorithmic issue we would like to address.

### 3.4.4 Rolling Horizon

At the beginning of a planning horizon, we know all the information about the patients and caregivers for the next $|\mathcal{P}|$ days. After running our algorithm, we get a home care scheduling plan for these $|\mathcal{P}|$ days. The first day of the plan, when $p = 1$, is actually implemented. One day later, there may be new requests appended and old requests deleted. We remove the information of the first day, update the information of the remaining $|\mathcal{P}| - 1$ days, and add the new $|\mathcal{P}|$th day's information to the end. In this way, we keep a fixed horizon length of $|\mathcal{P}|$ days, and repeat running our algorithm to solve the HCSP/WB for the next horizon. This idea of the rolling horizon originates from Bostel et al. (2008).

Due to our problem with the continuity restriction, we need to modify the original rolling horizon idea for our use. After one period of execution, any visit satisfied on the first day has its corresponding request locked to a certain caregiver; for the other visits of the same request later on, a restriction is placed in such a way that only that specific caregiver is allowed to perform the service.

Next, we present our numerical results.

## 3.5 NUMERICAL RESULTS

The parameters of our experiments are set as follows. Section 1.0.1 explains why the planning horizon $\mathcal{P}$ is set to 7 and how the average workload $L$ is calculated. In addition, the unit cost of traveling, $\mathcal{G}$, is \$0.275 per minute, because the agency compensates the caregivers \$0.55 per mile for fuel and also because caregivers are assumed to drive at a constant speed of 30 miles per hour. To help determine whether the HCA should purchase centrifuges for the caregivers, we need to know the unit cost of a centrifuge, $\mathcal{C}$. The fixed cost of a centrifuge is \$1000. We assume there is no variable cost associated with the operation of a centrifuge. We assume that the value of a centrifuge depreciates linearly over one year, and thus $\mathcal{C}$ is calculated as $\frac{1000*7}{366} \approx \$19.126$ per horizon.

In particular, the HCA is unsatisfied with the high fuel cost from February 3rd, 2012 to March 24th, 2012. Hence, we test our solution approach on these selected days. With the rolling horizon scheme, we run our algorithm once every day and get a scheduling plan for all caregivers. A total number of 51 days is selected, so we run our algorithm for 51 times. On February 3rd, 2012, the first day of the initial planning horizon, the HCA has $\mathcal{K} = 16$ caregivers and $\mathcal{J} = 6$ requests. As we roll the planning horizon over by one day, the size of set $\mathcal{J}$ grows bigger even though discharges can take place every day. Since patients are admitted for one episode, they are not discharged unless all of the requested visits are satisfied. After a succession of horizons is rolled over till the day when discharges outnumber new admits, the size of set $\mathcal{J}$ will then start decreasing. On the last day of our experiments, March 24th, 2012, $\mathcal{J}$ contains 261 requests in total. During the examined days, the average probability for a request to have blood test services is 10% according to the information provided by the HCA. Unfortunately, the HCA under our study does not have patients with synchronization or precedence requests among the examined dates. Hence, the probability of synchronized or sequential requests is 0. However, when patients with such requests arrive in the future, our model and solution approach are generic enough.

We implement our solution approach in C++. We run the code on an Intel Core 2 Duo E8400 PC with 2.94GB of memory. The C++ code interacts with the CPLEX optimizer, version 12.2[3], to solve the LP-relaxed MP of the column generation.

To better demonstrate the performance of our solution approach, we run the initialization algorithm which is basically a local search heuristic, the LP-based column generation algorithm which improves the output of the initialization algorithm, and the column generation algorithm with the rounding procedure, respectively. The computational results are summarized in Table 12. We report the following information: "Date" (the date), "Initial" (the objective of the heuristic), "Time" (the runtime of the heuristic in seconds), "CG" (the objective of the LP-based column generation), "CGTime" (the runtime of the LP-based column generation in seconds), "Iter" (the number of iterations), "Sub" (the number of subproblems solved), "Rounding" (the objective after LP solutions are rounded to integers), "Gap" (the difference in percentage between the column generation's rounded solution and the heuristic solution), "DailyCost" (the routing cost of the first day of each planning horizon), "noBlood" (the number of requests with blood test services), and "noRequest" (the number of requests). "Initial" is where the LP-based column generation starts the improvement endeavor and produces "CG", and "CG" is where the column generation algorithm with the rounding procedure starts searching for a feasible solution and produces "Rounding". Thus, "Initial" is no better than "Rounding", and "Rounding" is no better than "CG". However, only "Initial" and "Rounding" are guaranteed to be feasible solutions. We compare the performances of the heuristic with those of the column generation with the rounding procedure by calculating $\frac{\text{"Initial"} - \text{"Rounding"}}{\text{"Initial"}} * 100\%$, which is shown in "Gap". According to this column, if the column generation with the rounding procedure successfully finds any improvements,

[3]CPLEX is a trademark of IBM.

the differences in percentage are as significant as, if not more significant than, 8%. Nevertheless, the notable improvements are slightly compromised by the fact that improvements are not found on too many days. We also report the runtime of the heuristic in “Time” and that of the LP-based column generation in “CGTime”. All values in column “Time” are less than 1 second, and time in column “CGTime” is fast too even the longest experiment takes 1598.7 seconds on March 2nd, 2012. Because the difference in the runtime of the column generation with and without the rounding procedure is negligible, only one runtime is reported. Besides, we report the number of iterations in “Iter”. The number of subproblems are the product of the number of iterations and the number of subproblems in each iteration, that is, $|\mathcal{P}|$, which is shown in “Sub”. In addition, we report the corresponding cost of the first day of each experiment in “DailyCost”. The last two columns “noBlood” and “noRequest” contain the information of the problem sizes on each day, which shows a continually increasing trend.

Table 12: Results of a heuristic and the column generation without centrifuges.

| Date | Initial | Time | CG | CGTime | Iter | Sub | Rounding | Gap | DailyCost | noBlood | noRequest |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3-Feb | 195.6 | 0.0 | 129.3 | 3.0 | 4 | 28 | 129.3 | 34% | 122.4 | 1 | 6 |
| 4-Feb | 218.5 | 0.0 | 50.7 | 14.7 | 9 | 63 | 218.5 | 0% | 73.9 | 1 | 15 |
| 5-Feb | 174.4 | 0.0 | 169.7 | 1.8 | 4 | 28 | 174.4 | 0% | 26.6 | 1 | 16 |
| 6-Feb | 194.9 | 0.0 | 153.0 | 3.7 | 7 | 49 | 153.0 | 21% | 39.9 | 1 | 18 |
| 7-Feb | 321.0 | 0.0 | 156.0 | 17.1 | 13 | 91 | 321.0 | 0% | 117.0 | 1 | 24 |
| 8-Feb | 286.6 | 0.0 | 216.2 | 6.1 | 9 | 63 | 216.2 | 25% | 81.1 | 1 | 28 |
| 9-Feb | 282.3 | 0.0 | 150.3 | 34.8 | 19 | 133 | 160.0 | 43% | 29.0 | 1 | 37 |
| 10-Feb | 409.6 | 0.1 | 263.7 | 111.7 | 28 | 196 | 375.3 | 8% | 307.1 | 3 | 47 |
| 11-Feb | 299.8 | 0.1 | 177.8 | 69.4 | 20 | 140 | 299.8 | 0% | 269.6 | 4 | 53 |
| 12-Feb | 180.0 | 0.0 | 84.3 | 12.6 | 8 | 56 | 180.0 | 0% | 104.3 | 5 | 57 |
| 13-Feb | 253.7 | 0.1 | 129.8 | 48.0 | 15 | 105 | 253.7 | 0% | 179.3 | 5 | 64 |
| 14-Feb | 326.9 | 0.1 | 208.7 | 51.9 | 17 | 119 | 208.7 | 36% | 23.8 | 5 | 69 |
| 15-Feb | 361.7 | 0.1 | 212.3 | 138.8 | 26 | 182 | 361.7 | 0% | 351.5 | 5 | 76 |
| 16-Feb | 217.1 | 0.1 | 109.6 | 100.0 | 22 | 154 | 217.1 | 0% | 162.0 | 5 | 81 |
| 17-Feb | 455.8 | 0.1 | 395.9 | 397.7 | 39 | 273 | 455.8 | 0% | 357.4 | 6 | 88 |
| 18-Feb | 508.6 | 0.1 | 432.8 | 108.4 | 22 | 154 | 508.6 | 0% | 360.7 | 6 | 93 |
| 19-Feb | 331.1 | 0.1 | 260.5 | 43.8 | 15 | 105 | 331.1 | 0% | 214.7 | 7 | 95 |
| 20-Feb | 277.3 | 0.1 | 227.2 | 45.2 | 16 | 112 | 277.3 | 0% | 185.8 | 7 | 98 |
| 21-Feb | 345.6 | 0.1 | 293.8 | 71.5 | 19 | 133 | 345.6 | 0% | 243.3 | 8 | 102 |
| 22-Feb | 376.3 | 0.1 | 317.7 | 57.3 | 15 | 105 | 376.3 | 0% | 317.0 | 9 | 107 |
| 23-Feb | 452.7 | 0.1 | 297.6 | 493.7 | 39 | 273 | 452.7 | 0% | 356.2 | 9 | 114 |
| 24-Feb | 555.9 | 0.2 | 368.5 | 818.4 | 48 | 336 | 555.9 | 0% | 358.7 | 11 | 122 |
| 25-Feb | 519.7 | 0.1 | 270.5 | 1225.6 | 54 | 378 | 280.6 | 46% | 0.5 | 12 | 129 |
| 26-Feb | 672.1 | 0.2 | 421.9 | 891.0 | 67 | 469 | 672.1 | 0% | 382.8 | 12 | 132 |
| 27-Feb | 535.6 | 0.1 | 318.2 | 175.6 | 25 | 175 | 535.6 | 0% | 185.7 | 13 | 134 |
| 28-Feb | 743.1 | 0.1 | 373.6 | 379.0 | 36 | 252 | 743.1 | 0% | 339.8 | 13 | 138 |
| 29-Feb | 558.4 | 0.1 | 367.4 | 177.0 | 27 | 189 | 558.4 | 0% | 250.3 | 13 | 141 |
| 1-Mar | 525.2 | 0.2 | 468.1 | 584.2 | 50 | 350 | 525.2 | 0% | 296.0 | 13 | 146 |
| 2-Mar | 592.3 | 0.1 | 383.5 | 1681.0 | 63 | 441 | 592.3 | 0% | 392.8 | 13 | 153 |
| 3-Mar | 333.2 | 0.1 | 301.1 | 1673.9 | 77 | 539 | 304.9 | 9% | 236.4 | 13 | 162 |
| 4-Mar | 288.6 | 0.1 | 183.1 | 44.6 | 11 | 77 | 288.6 | 0% | 144.2 | 14 | 165 |
| 5-Mar | 326.3 | 0.1 | 194.9 | 134.9 | 20 | 140 | 326.3 | 0% | 199.6 | 14 | 171 |
| 6-Mar | 411.5 | 0.1 | 402.4 | 67.0 | 16 | 112 | 411.5 | 0% | 322.1 | 14 | 172 |
| 7-Mar | 609.0 | 0.1 | 277.5 | 169.4 | 21 | 147 | 609.0 | 0% | 297.7 | 16 | 178 |
| 8-Mar | 669.1 | 0.1 | 612.7 | 717.0 | 42 | 294 | 669.1 | 0% | 406.0 | 16 | 183 |
| 9-Mar | 651.0 | 0.2 | 544.8 | 1478.4 | 51 | 357 | 651.0 | 0% | 405.1 | 17 | 194 |
| 10-Mar | 532.6 | 0.1 | 405.3 | 535.7 | 36 | 252 | 532.6 | 0% | 381.9 | 19 | 198 |
| 11-Mar | 283.5 | 0.1 | 283.5 | 60.8 | 14 | 98 | 283.5 | 0% | 173.1 | 19 | 201 |
| 12-Mar | 253.8 | 0.1 | 227.4 | 43.0 | 12 | 84 | 227.4 | 10% | 224.1 | 19 | 202 |
| 13-Mar | 533.4 | 0.1 | 511.5 | 222.8 | 25 | 175 | 533.4 | 0% | 342.3 | 20 | 207 |
| 14-Mar | 401.5 | 0.1 | 401.5 | 64.6 | 15 | 105 | 401.5 | 0% | 289.3 | 20 | 209 |
| 15-Mar | 491.9 | 0.2 | 491.9 | 145.4 | 22 | 154 | 491.9 | 0% | 360.4 | 21 | 211 |
| 16-Mar | 589.5 | 0.2 | 533.1 | 501.2 | 34 | 238 | 589.5 | 0% | 440.2 | 22 | 216 |
| 17-Mar | 536.3 | 0.2 | 489.0 | 1009.8 | 47 | 329 | 536.3 | 0% | 390.9 | 22 | 219 |
| 18-Mar | 427.7 | 0.2 | 264.0 | 294.8 | 27 | 189 | 427.7 | 0% | 302.7 | 23 | 224 |
| 19-Mar | 591.4 | 0.2 | 337.1 | 673.4 | 35 | 245 | 591.4 | 0% | 362.9 | 24 | 230 |
| 20-Mar | 815.2 | 0.2 | 621.4 | 551.8 | 31 | 217 | 815.2 | 0% | 459.7 | 24 | 238 |
| 21-Mar | 639.6 | 0.2 | 498.2 | 789.9 | 39 | 273 | 639.6 | 0% | 438.3 | 24 | 243 |
| 22-Mar | 1306.5 | 0.2 | 1238.9 | 1333.6 | 48 | 336 | 1306.5 | 0% | 361.2 | 24 | 249 |
| 23-Mar | 795.8 | 0.2 | 795.8 | 561.5 | 35 | 245 | 795.8 | 0% | 523.4 | 25 | 255 |
| 24-Mar | 632.7 | 0.2 | 596.9 | 632.0 | 37 | 259 | 632.7 | 0% | 443.7 | 26 | 261 |

It is worth noting that on March 20th, 23rd, and 24th, 2012, the column generation runs out of memory due to the size of the problem. Thus, on these days, we only report the objectives we obtain when the program is terminated due to the memory depletion.

In addition to solving the original HCSP/WB, we also solve the model with a variation when caregivers are equipped with centrifuges to perform blood tests on-site. We only need to slightly modify the original solution approach by setting all requests to contain no blood test services; the rest of the solution algorithm remains the same. We run the modified solution approach for the same selected days with the rolling horizon scheme. The results are summarized in Table 13. Since the number of requests with blood test services is set to 0, we delete the column of "noBlood" and keep the remaining columns as shown in Table 12. We name the columns in the same way as we do in Table 12 and find similar results. In addition, we compare the costs of solutions with and without centrifuges and introduce two new columns: "DifRound" (the cost differences in the column of "Rounding" between the solutions with centrifuges and the ones without centrifuges) and "DifDaily" (the daily cost differences in the column of "DailyCost" between the solutions with centrifuges and the ones without centrifuges). Both columns contain both negative and positive values, and the total cost difference is \$822.5 and the total daily cost difference is \$355.5 by summing up each column. Thus, centrifuges save the HCA \$355.5 for the selected 51 days. The cost of equipping a centrifuge for each caregiver for the same length is $\$\mathcal{C} * 51/7 * 16 \approx \$2229.5$. Therefore, with 10% requests with blood test services, the investment cost of centrifuges exceeds the total savings. Therefore, the HCA should not make the investment.

Table 13: Results of a heuristic and the column generation with centrifuges.

| Date | Initial | Time | CG | CGTime | Iter | Sub | Rounding | Gap | DifRound | DailyCost | DifDaily | noRequ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3-Feb | 194.7 | 0.0 | 128.3 | 3.0 | 4 | 28 | 128.3 | 34% | 0.9 | 121.5 | 0.9 | 6 |
| 4-Feb | 218.5 | 0.0 | 50.7 | 15.1 | 9 | 63 | 218.5 | 0% | 0.0 | 73.9 | 0.0 | 15 |
| 5-Feb | 174.4 | 0.0 | 169.7 | 1.6 | 4 | 28 | 174.4 | 0% | 0.0 | 26.6 | 0.0 | 16 |
| 6-Feb | 194.9 | 0.0 | 153.0 | 3.7 | 7 | 49 | 153.0 | 21% | 0.0 | 39.9 | 0.0 | 18 |
| 7-Feb | 321.0 | 0.0 | 156.0 | 17.3 | 13 | 91 | 321.0 | 0% | 0.0 | 117.0 | 0.0 | 24 |
| 8-Feb | 286.6 | 0.0 | 216.2 | 5.8 | 9 | 63 | 216.2 | 25% | 0.0 | 81.1 | 0.0 | 28 |
| 9-Feb | 282.3 | 0.0 | 150.3 | 35.3 | 19 | 133 | 160.0 | 43% | 0.0 | 29.0 | 0.0 | 37 |
| 10-Feb | 381.0 | 0.1 | 219.3 | 217.8 | 41 | 287 | 346.8 | 9% | 28.5 | 280.0 | 27.1 | 47 |
| 11-Feb | 295.4 | 0.1 | 169.9 | 69.3 | 20 | 140 | 295.4 | 0% | 4.4 | 265.1 | 4.4 | 53 |
| 12-Feb | 176.0 | 0.0 | 85.5 | 12.3 | 8 | 56 | 176.0 | 0% | 4.0 | 102.3 | 2.0 | 57 |
| 13-Feb | 251.7 | 0.1 | 127.8 | 48.5 | 15 | 105 | 251.7 | 0% | 2.0 | 177.3 | 2.0 | 64 |
| 14-Feb | 326.9 | 0.1 | 208.7 | 53.0 | 17 | 119 | 208.7 | 36% | 0.0 | 23.8 | 0.0 | 69 |
| 15-Feb | 361.7 | 0.1 | 212.3 | 130.6 | 26 | 182 | 361.7 | 0% | 0.0 | 351.5 | 0.0 | 76 |
| 16-Feb | 217.1 | 0.1 | 109.6 | 100.7 | 22 | 154 | 217.1 | 0% | 0.0 | 162.0 | 0.0 | 81 |
| 17-Feb | 427.4 | 0.1 | 365.8 | 356.5 | 37 | 259 | 427.4 | 0% | 28.4 | 329.0 | 28.4 | 88 |
| 18-Feb | 502.6 | 0.1 | 424.8 | 120.1 | 23 | 161 | 502.6 | 0% | 6.0 | 357.7 | 3.0 | 93 |
| 19-Feb | 319.8 | 0.1 | 218.9 | 42.7 | 15 | 105 | 319.8 | 0% | 11.3 | 208.1 | 6.6 | 95 |
| 20-Feb | 272.5 | 0.1 | 221.8 | 44.7 | 16 | 112 | 272.5 | 0% | 4.8 | 184.2 | 1.6 | 98 |
| 21-Feb | 342.5 | 0.1 | 290.6 | 71.8 | 19 | 133 | 342.5 | 0% | 3.2 | 241.7 | 1.6 | 102 |
| 22-Feb | 374.8 | 0.1 | 304.1 | 78.4 | 18 | 126 | 374.8 | 0% | 1.5 | 315.5 | 1.5 | 107 |
| 23-Feb | 451.1 | 0.1 | 301.1 | 518.1 | 40 | 280 | 451.1 | 0% | 1.6 | 354.6 | 1.6 | 114 |
| 24-Feb | 547.8 | 0.2 | 366.3 | 1112.8 | 57 | 399 | 547.8 | 0% | 8.1 | 350.5 | 8.1 | 122 |
| 25-Feb | 513.9 | 0.1 | 275.5 | 540.1 | 40 | 280 | 281.2 | 45% | -0.6 | 0.5 | 0.0 | 129 |
| 26-Feb | 664.5 | 0.2 | 414.6 | 1377.7 | 84 | 588 | 664.5 | 0% | 7.6 | 375.2 | 7.6 | 132 |
| 27-Feb | 518.7 | 0.1 | 318.3 | 117.3 | 19 | 133 | 518.7 | 0% | 16.8 | 177.2 | 8.4 | 134 |
| 28-Feb | 734.7 | 0.1 | 369.6 | 443.0 | 39 | 273 | 734.7 | 0% | 8.4 | 331.4 | 8.4 | 138 |
| 29-Feb | 557.5 | 0.1 | 365.2 | 222.4 | 31 | 217 | 557.5 | 0% | 0.9 | 250.4 | -0.1 | 141 |
| 1-Mar | 518.3 | 0.1 | 460.8 | 608.5 | 50 | 350 | 518.3 | 0% | 6.9 | 289.0 | 6.9 | 146 |
| 2-Mar | 585.6 | 0.1 | 377.3 | 1598.7 | 61 | 427 | 585.6 | 0% | 6.6 | 386.1 | 6.6 | 153 |
| 3-Mar | 333.2 | 0.1 | 301.1 | 778.2 | 47 | 329 | 304.9 | 9% | 0.0 | 249.8 | -13.4 | 162 |
| 4-Mar | 288.6 | 0.1 | 183.1 | 47.6 | 12 | 84 | 288.6 | 0% | 0.0 | 158.5 | -14.3 | 165 |
| 5-Mar | 326.3 | 0.1 | 194.9 | 85.9 | 16 | 112 | 326.3 | 0% | 0.0 | 213.2 | -13.5 | 171 |
| 6-Mar | 411.5 | 0.1 | 402.4 | 126.7 | 23 | 161 | 411.5 | 0% | 0.0 | 319.1 | 3.0 | 172 |
| 7-Mar | 609.0 | 0.1 | 277.5 | 189.2 | 22 | 154 | 609.0 | 0% | 0.0 | 311.7 | -14.0 | 178 |
| 8-Mar | 669.1 | 0.1 | 612.7 | 558.7 | 37 | 259 | 669.1 | 0% | 0.0 | 400.9 | 5.0 | 183 |
| 9-Mar | 651.0 | 0.2 | 544.8 | 1202.5 | 47 | 329 | 651.0 | 0% | 0.0 | 397.7 | 7.4 | 194 |
| 10-Mar | 532.6 | 0.2 | 405.3 | 679.1 | 41 | 287 | 532.6 | 0% | 0.0 | 372.4 | 9.5 | 198 |
| 11-Mar | 283.5 | 0.1 | 283.5 | 62.1 | 14 | 98 | 283.5 | 0% | 0.0 | 190.8 | -17.7 | 201 |
| 12-Mar | 253.8 | 0.1 | 227.4 | 57.5 | 14 | 98 | 227.4 | 10% | 0.0 | 57.0 | 167.1 | 202 |
| 13-Mar | 533.4 | 0.1 | 511.5 | 228.7 | 27 | 189 | 533.4 | 0% | 0.0 | 374.0 | -31.7 | 207 |
| 14-Mar | 401.5 | 0.1 | 401.5 | 80.2 | 17 | 119 | 401.5 | 0% | 0.0 | 295.6 | -6.3 | 209 |
| 15-Mar | 491.9 | 0.2 | 491.9 | 125.0 | 20 | 140 | 491.9 | 0% | 0.0 | 355.9 | 4.5 | 211 |
| 16-Mar | 589.5 | 0.2 | 533.1 | 773.1 | 42 | 294 | 589.5 | 0% | 0.0 | 426.4 | 13.8 | 216 |
| 17-Mar | 536.3 | 0.2 | 489.0 | 1071.4 | 49 | 343 | 536.3 | 0% | 0.0 | 376.4 | 14.5 | 219 |
| 18-Mar | 427.7 | 0.2 | 264.0 | 336.3 | 29 | 203 | 427.7 | 0% | 0.0 | 308.5 | -5.8 | 224 |
| 19-Mar | 591.4 | 0.2 | 337.1 | 668.3 | 35 | 245 | 591.4 | 0% | 0.0 | 345.3 | 17.5 | 230 |
| 20-Mar | 815.2 | 0.2 | 621.4 | 551.8 | 31 | 217 | 815.2 | 0% | 0.0 | 459.7 | 0.0 | 238 |
| 21-Mar | 639.6 | 0.2 | 498.2 | 789.9 | 39 | 273 | 639.6 | 0% | 0.0 | 438.3 | 0.0 | 243 |
| 22-Mar | 690.8 | 0.1 | 498.9 | 78.4 | 18 | 126 | 690.8 | 0% | 615.7 | 315.5 | 45.7 | 249 |
| 23-Mar | 751.3 | 0.2 | 751.3 | 3.2 | 1 | 7 | 751.3 | 0% | 44.5 | 483.0 | 40.3 | 255 |
| 24-Mar | 621.7 | 0.2 | 558.3 | 1215.6 | 50 | 350 | 621.7 | 0% | 11.0 | 426.3 | 17.4 | 261 |

In these experiments, the column generation also encounters memory issue on March 20th and 23rd, 2012. The same memory resolution technique is applied here as explained before.

Moreover, we compare our solution with the current practice on the basis of the weekly fuel cost. The results are summarized in Table 14. It contains the following columns: "RealWeeklyCost" (the real weekly fuel cost), "CGWeeklyCost"(the weekly fuel cost for the column generation with the rounding procedure), "Dif" (the difference between "RealWeeklyCost" and "CGWeeklyCost"), and "Gap" ($\frac{\text{"Dif"}}{\text{"RealWeeklyCost"}} * 100\%$). When centrifuges are not available, the real costs are higher than our costs in the first three weeks, but the real costs are lower in the remaining five weeks. When centrifuges are available, the real costs are higher than our costs in the first four weeks, but the real costs are lower in the remaining four weeks. Even though the HCA's current practice yields lower costs than our solution in the later weeks, it cannot satisfy all the constraints that it defines, especially the workload balance constraints. This HCA has some caregivers with workloads of more than 40 patients, while some others have much lighter workloads. This leads to a lower level of employee satisfaction, which is harmful to the employees' productivity and the service quality. On the contrary, our solution satisfies all of the constraints with acceptable higher cost.

Table 14: Weekly cost comparison of the column generation with the current practice.

| | Without Centrifuges | | | With Centrifuges | | | |
|---|---|---|---|---|---|---|---|
| | CGWeeklyCost | Dif | Gap | CGWeeklyCost | Dif | Gap | RealWeeklyCost |
| 4-Feb | 196.3 | 58.0 | 22.8% | 195.4 | 58.9 | 23.2% | 254.3 |
| 11-Feb | 870.4 | 19.6 | 2.2% | 838.8 | 51.1 | 5.7% | 889.9 |
| 18-Feb | 1538.8 | 129.0 | 7.7% | 1503.5 | 164.4 | 9.9% | 1667.9 |
| 25-Feb | 1676.1 | -8.3 | -0.5% | 1655.2 | 12.7 | 0.8% | 1667.9 |
| 3-Mar | 2083.7 | -622.1 | -42.6% | 2059.2 | -597.6 | -40.9% | 1461.6 |
| 10-Mar | 2156.6 | -695.0 | -47.5% | 2173.5 | -711.9 | -48.7% | 1461.6 |
| 17-Mar | 2220.2 | -558.1 | -33.6% | 2076.2 | -414.1 | -24.9% | 1662.1 |
| 24-Mar | 2891.7 | -1229.6 | -74.0% | 2776.5 | -1114.4 | -67.1% | 1662.1 |

To help the HCA evaluate our solution, we report the weekly workloads of caregivers when centrifuges are available and not available. The results are summarized in Table 15, where "c1" stands for caregiver number 1 and the other titles follow the same naming convention. When the dates get later, the HCA has more requests and thus each caregiver's assigned number of requests grows bigger. When centrifuges are not available, the caregiver with the heaviest workload is caregiver number 1, who is assigned to 24 requests; when centrifuges are available, the caregiver with the heaviest workload is also caregiver number 1, who is assigned to 25 requests. The heaviest workload by our solution is much lower than that of the HCA's current practice, and thus our solution leads to more equal workloads among the caregivers. Our solution helps the HCA achieve better employee satisfaction, eventually translating into higher productivity and better service quality.

Table 15: Weekly workloads of caregivers with and without centrifuges.

| | Date | c1 | c2 | c3 | c4 | c5 | c6 | c7 | c8 | c9 | c10 | c11 | c12 | c13 | c14 | c15 | c16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4-Feb | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| | 11-Feb | 3 | 2 | 4 | 3 | 4 | 8 | 2 | 2 | 2 | 2 | 3 | 2 | 1 | 6 | 1 | 1 |
| | 18-Feb | 9 | 7 | 9 | 4 | 5 | 10 | 5 | 5 | 3 | 4 | 4 | 3 | 1 | 7 | 3 | 2 |
| Without | 25-Feb | 10 | 10 | 9 | 8 | 7 | 11 | 8 | 7 | 7 | 5 | 6 | 5 | 2 | 7 | 5 | 3 |
| Centrifuges | 3-Mar | 14 | 12 | 12 | 10 | 12 | 12 | 8 | 10 | 7 | 6 | 9 | 5 | 7 | 11 | 6 | 3 |
| | 10-Mar | 17 | 14 | 15 | 13 | 13 | 13 | 13 | 13 | 8 | 9 | 12 | 9 | 8 | 10 | 9 | 5 |
| | 17-Mar | 20 | 17 | 15 | 14 | 14 | 16 | 14 | 15 | 10 | 9 | 12 | 10 | 8 | 10 | 10 | 5 |
| | 24-Mar | 24 | 21 | 18 | 18 | 17 | 18 | 14 | 18 | 13 | 11 | 15 | 11 | 12 | 15 | 13 | 9 |
| | 4-Feb | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| | 11-Feb | 3 | 2 | 4 | 3 | 4 | 8 | 2 | 2 | 2 | 2 | 3 | 2 | 1 | 6 | 1 | 1 |
| | 18-Feb | 8 | 6 | 8 | 4 | 5 | 10 | 5 | 5 | 3 | 4 | 4 | 3 | 2 | 7 | 2 | 2 |
| With | 25-Feb | 10 | 10 | 10 | 7 | 7 | 11 | 8 | 7 | 8 | 5 | 5 | 5 | 2 | 7 | 5 | 3 |
| Centrifuges | 3-Mar | 12 | 12 | 13 | 9 | 12 | 12 | 8 | 10 | 8 | 5 | 8 | 5 | 7 | 11 | 8 | 3 |
| | 10-Mar | 16 | 13 | 16 | 13 | 13 | 14 | 13 | 12 | 9 | 9 | 11 | 9 | 8 | 10 | 9 | 6 |
| | 17-Mar | 20 | 16 | 17 | 14 | 13 | 15 | 14 | 14 | 11 | 10 | 12 | 9 | 8 | 10 | 9 | 7 |
| | 24-Mar | 25 | 19 | 21 | 18 | 17 | 17 | 14 | 17 | 14 | 11 | 15 | 12 | 11 | 14 | 11 | 11 |

Moreover, since the HCA violates the workload constraints, we relax them and solve the 51 problems again. We modify the solution approach to reflect the relaxation by assigning requests to the closest caregiver instead of the lightest-loaded caregiver. We present the results in Table 16.

Table 16: Results of a heuristic with relaxed workload constraints and the CG without centrifuges.

| Date | Initial | Time | CG | CGTime | Iter | Sub | Rounding | Gap | DailyCost | noBlood | noRequest |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3-Feb | 157.9 | 0.0 | 136.7 | 3.7 | 7 | 49 | 136.7 | 13% | 122.1 | 1 | 6 |
| 4-Feb | 91.9 | 0.0 | 57.0 | 13.5 | 10 | 70 | 91.9 | 0% | 31.5 | 1 | 15 |
| 5-Feb | 84.0 | 0.0 | 81.2 | 0.9 | 4 | 28 | 81.2 | 3% | 4.0 | 1 | 16 |
| 6-Feb | 87.5 | 0.0 | 83.9 | 1.7 | 6 | 42 | 84.0 | 4% | 13.2 | 1 | 18 |
| 7-Feb | 116.1 | 0.0 | 116.1 | 5.9 | 9 | 63 | 116.1 | 0% | 53.3 | 1 | 24 |
| 8-Feb | 74.4 | 0.0 | 74.4 | 3.6 | 9 | 63 | 74.4 | 0% | 43.1 | 1 | 28 |
| 9-Feb | 52.7 | 0.0 | 52.7 | 10.5 | 12 | 84 | 52.7 | 0% | 42.9 | 1 | 37 |
| 10-Feb | 142.3 | 0.0 | 142.3 | 23.3 | 15 | 105 | 142.3 | 0% | 133.8 | 3 | 47 |
| 11-Feb | 65.2 | 0.0 | 65.2 | 31.6 | 15 | 105 | 65.2 | 0% | 56.0 | 4 | 53 |
| 12-Feb | 59.0 | 0.0 | 59.0 | 8.5 | 8 | 56 | 59.0 | 0% | 36.0 | 5 | 57 |
| 13-Feb | 66.0 | 0.1 | 66.0 | 14.2 | 9 | 63 | 66.0 | 0% | 41.8 | 5 | 64 |
| 14-Feb | 76.2 | 0.1 | 76.2 | 39.9 | 17 | 119 | 76.2 | 0% | 65.5 | 5 | 69 |
| 15-Feb | 43.5 | 0.1 | 43.5 | 20.3 | 11 | 77 | 43.5 | 0% | 38.0 | 5 | 76 |
| 16-Feb | 63.1 | 0.1 | 63.1 | 75.3 | 22 | 154 | 63.1 | 0% | 48.0 | 5 | 81 |
| 17-Feb | 216.9 | 0.1 | 216.9 | 84.0 | 22 | 154 | 216.9 | 0% | 174.6 | 6 | 88 |
| 18-Feb | 123.6 | 0.1 | 123.6 | 46.1 | 17 | 119 | 123.6 | 0% | 90.8 | 6 | 93 |
| 19-Feb | 119.2 | 0.1 | 116.5 | 38.1 | 16 | 112 | 119.2 | 0% | 65.5 | 7 | 95 |
| 20-Feb | 94.1 | 0.1 | 92.8 | 31.9 | 15 | 105 | 94.1 | 0% | 54.6 | 7 | 98 |
| 21-Feb | 142.8 | 0.1 | 142.8 | 50.1 | 18 | 126 | 142.8 | 0% | 104.0 | 8 | 102 |
| 22-Feb | 114.6 | 0.1 | 114.6 | 38.6 | 14 | 98 | 114.6 | 0% | 97.8 | 9 | 107 |
| 23-Feb | 133.1 | 0.1 | 133.1 | 276.2 | 34 | 238 | 133.1 | 0% | 93.8 | 9 | 114 |
| 24-Feb | 276.8 | 0.2 | 253.9 | 935.8 | 58 | 406 | 276.8 | 0% | 113.2 | 11 | 122 |
| 25-Feb | 223.7 | 0.2 | 206.9 | 358.8 | 42 | 294 | 223.7 | 0% | 82.3 | 12 | 129 |
| 26-Feb | 295.2 | 0.1 | 272.8 | 40.8 | 16 | 112 | 277.6 | 6% | 0.5 | 12 | 132 |
| 27-Feb | 398.2 | 0.1 | 311.9 | 315.0 | 40 | 280 | 398.2 | 0% | 86.5 | 13 | 134 |
| 28-Feb | 397.1 | 0.2 | 342.4 | 238.8 | 32 | 224 | 397.1 | 0% | 112.1 | 13 | 138 |
| 29-Feb | 352.0 | 0.1 | 333.0 | 241.1 | 38 | 266 | 352.0 | 0% | 68.0 | 13 | 141 |
| 1-Mar | 369.0 | 0.2 | 344.0 | 591.4 | 56 | 392 | 369.0 | 0% | 203.3 | 13 | 146 |
| 2-Mar | 279.8 | 0.3 | 267.9 | 717.5 | 48 | 336 | 279.8 | 0% | 118.2 | 13 | 152 |
| 3-Mar | 235.1 | 0.2 | 230.5 | 839.6 | 46 | 322 | 235.1 | 0% | 193.7 | 13 | 162 |
| 4-Mar | 91.0 | 0.1 | 88.4 | 23.0 | 9 | 63 | 90.2 | 1% | 2.6 | 14 | 165 |
| 5-Mar | 141.2 | 0.2 | 141.2 | 62.3 | 16 | 112 | 141.2 | 0% | 90.6 | 14 | 171 |
| 6-Mar | 127.2 | 0.2 | 127.2 | 66.8 | 19 | 133 | 127.2 | 0% | 102.4 | 14 | 172 |
| 7-Mar | 111.4 | 0.1 | 111.4 | 86.8 | 18 | 126 | 111.4 | 0% | 73.8 | 16 | 178 |
| 8-Mar | 159.1 | 0.2 | 159.1 | 189.5 | 26 | 182 | 159.1 | 0% | 108.9 | 16 | 183 |
| 9-Mar | 193.0 | 0.2 | 193.0 | 839.8 | 41 | 287 | 193.0 | 0% | 127.8 | 17 | 194 |
| 10-Mar | 181.6 | 0.2 | 150.4 | 949.8 | 46 | 322 | 181.6 | 0% | 131.7 | 19 | 198 |
| 11-Mar | 71.9 | 0.2 | 71.9 | 53.9 | 15 | 105 | 71.9 | 0% | 57.0 | 19 | 201 |
| 12-Mar | 72.2 | 0.1 | 72.2 | 34.0 | 12 | 84 | 72.2 | 0% | 63.4 | 19 | 202 |
| 13-Mar | 98.4 | 0.2 | 98.4 | 145.7 | 21 | 147 | 98.4 | 0% | 77.0 | 20 | 207 |
| 14-Mar | 79.5 | 0.3 | 79.5 | 71.4 | 16 | 112 | 79.5 | 0% | 63.5 | 20 | 209 |
| 15-Mar | 148.6 | 0.2 | 148.6 | 102.3 | 18 | 126 | 148.6 | 0% | 92.1 | 21 | 211 |
| 16-Mar | 262.8 | 0.2 | 254.2 | 1652.0 | 54 | 378 | 262.8 | 0% | 171.9 | 22 | 216 |
| 17-Mar | 225.4 | 0.2 | 225.4 | 788.1 | 41 | 287 | 225.4 | 0% | 157.3 | 22 | 219 |
| 18-Mar | 101.8 | 0.1 | 101.8 | 178.7 | 22 | 154 | 101.8 | 0% | 72.5 | 23 | 224 |
| 19-Mar | 329.0 | 0.5 | 329.0 | 230.9 | 22 | 154 | 329.0 | 0% | 178.1 | 24 | 230 |
| 20-Mar | 319.6 | 0.3 | 319.6 | 267.7 | 26 | 182 | 319.6 | 0% | 238.7 | 24 | 238 |
| 21-Mar | 184.1 | 0.2 | 184.1 | 376.5 | 33 | 231 | 184.1 | 0% | 135.9 | 24 | 243 |
| 22-Mar | 201.4 | 0.2 | 199.2 | 974.4 | 49 | 343 | 201.4 | 0% | 126.2 | 24 | 249 |
| 23-Mar | 250.2 | 0.3 | 250.2 | 331.6 | 30 | 210 | 250.2 | 0% | 168.0 | 25 | 255 |
| 24-Mar | 244.8 | 0.3 | 244.8 | 315.9 | 30 | 210 | 244.8 | 0% | 158.9 | 26 | 261 |

Similar to the analysis conducted before, we compare the weekly cost in our solution with the real cost, whose results are summarized in Table 17. We are able to reduce costs each week by a large percentage. In total, our solution costs $4786.5 while the HCA pays $10727.4 for fuel. If we ignore the workload constraints as the current practice does, our solution produces much lower cost.

Table 17: Weekly cost comparison of the CG with relaxed workload constraints with the current practice.

| | Without Centrifuges | | | |
|---|---|---|---|---|
| | CGWeeklyCost | Dif | Gap | RealWeeklyCost |
| 4-Feb | 153.6 | 100.7 | 39.6% | 254.3 |
| 11-Feb | 346.3 | 543.6 | 61.1% | 889.9 |
| 18-Feb | 494.6 | 1173.2 | 70.3% | 1667.9 |
| 25-Feb | 611.2 | 1056.7 | 63.4% | 1667.9 |
| 3-Mar | 782.4 | 679.2 | 46.5% | 1461.6 |
| 10-Mar | 637.9 | 823.8 | 56.4% | 1461.6 |
| 17-Mar | 682.2 | 979.9 | 59.0% | 1662.1 |
| 24-Mar | 1078.3 | 583.8 | 35.1% | 1662.1 |

Additionally, we report the weekly workloads in Table 18. The heaviest workload is 72 patients by caregiver 11 and the lightest workload is 0 patient by multiple caregivers.

Table 18: Weekly workloads of caregivers with relaxed workload constraints without centrifuges.

| | c1 | c2 | c3 | c4 | c5 | c6 | c7 | c8 | c9 | c10 | c11 | c12 | c13 | c14 | c15 | c16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4-Feb | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 0 | 0 |
| 11-Feb | 4 | 0 | 5 | 1 | 0 | 1 | 2 | 0 | 1 | 0 | 9 | 1 | 0 | 15 | 1 | 0 |
| 18-Feb | 12 | 0 | 5 | 4 | 0 | 2 | 2 | 0 | 2 | 0 | 26 | 1 | 0 | 20 | 4 | 0 |
| 25-Feb | 19 | 1 | 6 | 4 | 0 | 2 | 2 | 2 | 2 | 0 | 40 | 1 | 0 | 21 | 5 | 0 |
| 3-Mar | 29 | 2 | 7 | 8 | 0 | 2 | 2 | 2 | 3 | 0 | 49 | 1 | 0 | 24 | 6 | 0 |
| 10-Mar | 29 | 2 | 14 | 12 | 0 | 2 | 4 | 2 | 3 | 0 | 58 | 1 | 0 | 30 | 6 | 0 |
| 17-Mar | 49 | 2 | 17 | 13 | 0 | 3 | 4 | 2 | 4 | 0 | 60 | 1 | 0 | 34 | 6 | 0 |
| 24-Mar | 62 | 5 | 23 | 15 | 1 | 5 | 4 | 2 | 4 | 0 | 72 | 0 | 0 | 41 | 7 | 0 |

## 3.6 CONCLUSIONS

In this paper, we address the problem of assigning caregivers of different qualifications to serve patients with different requests, subject to a host of practical constraints. We propose a novel mixed integer linear model with multiple periods to solve an home care scheduling problem with workload balance, namely the HCSB/WB. We apply the column generation technique to decompose the problem to its subproblems, each of which is solvable period by period so that the problem size is reduced, and solve them iteratively. We analyze the performances of our model and solution approach using real-life data from an HCA in Tuscaloosa, and demonstrate improvement by our approach over the current practice. We offer suggestions and insights for the HCA to improve its scheduling efficiency.

There are three major contributions of our paper. First, we offer a more comprehensive perspective than previous works by incorporating a host of practical constraints. Second, we are the first to present a formal, mathematical model for the problem of HCSP with workload balance. Third, we are the first to apply column generation to successfully solve the HCSP/WB under a rolling horizon scheme.

This paper can be extended in the following three directions. First, we will relax the assumption that the HCA only approves caregivers of unexpected asking for a leave at least one day ahead of time. If the HCA approves absence with a shorter notice, the central dispatcher needs to find a backup nurse to carry out the absent caregiver's tasks. As a result, the continuity constraints will not hold. Second, we will apply some metaheuristics on the neighborhood search algorithms to solve the subproblem faster. Third, we will improve the solution quality by combining the column generation with the branch-and-bound technique, on the condition that the runtime at each node is shortened.

## REFERENCES


Barnhartn, C., E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, P. H. Vance. 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations Research* **46**(3) 316–329.

Begur, S. V., D. M. Miller, J. R. Weaver. 1997. An integrated spatial DSS for scheduling and routing home-health-care nurses. *Interfaces* **27**(4) 35–48.

Bostel, N., P. Dejax, P. Guez, F. Tricoire. 2008. Multi-period planning and routing on a rolling horizon for field force optimization logistics. *The Vehicle Routing Problem: Latest Advances and New Challenges* 503–525.

Bräysy, O., M. Gendreau. 2005. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science* **39**(1) 104–118.

Bredström, D., M. Rönnqvist. 2008. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research* **191**(1) 19–31.

Dantzig, G. B., P. Wolfe. 1960. Decomposition principle for linear programs. *Operations Research* **8**(1) 101–111.

Desrosiers, J., Y. Dumas, M. M. Solomon, F. Soumis. 1995. Time constrained routing and scheduling. *Handbooks in Operations Research and Management Science* **8** 35–139.

Eveborn, P., P. Flisberg, M. Ronnqvist. 2006. Laps care–an operational system for staff planning of home care. *European Journal of Operational Research* **171**(3) 962–976.

Gaudioso, M., G. Paletta. 1992. A heuristic for the periodic vehicle routing problem. *Transportation Science* **26**(2) 86–92.

National Association for Home Care. 2008. Escalating energy costs threaten health care for critically ill and homebound seniors. *Washington: NHCA*.

National Association for Home Care. 2010. Basic statistics about home care. *Washington: NHCA*. 1–14

Nickel, S., M. Schröder, J. Steeg. 2012. Mid-Term and short-term planning support for home health care services. *European Journal of Operational Research* **219**(3) 574–587.

Rasmussen, M. S., T. Justesen, A. Dohn, J. Larsen. 2012. The home care crew scheduling problem: preference-based visit clustering and temporal dependencies. *European Journal of Operational Research* **219**(3) 598–610.

## APPENDIX

## FULL MODEL

$$\text{Minimize} \sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}\setminus i}\mathcal{G}t_{ij}X_{pkij}.$$

$$X_{pkij} \le U_{kj} \le e_{kj}, \quad \forall p\in\mathcal{P},\ i\in\mathcal{V},\ j\in\mathcal{J},\ k\in\mathcal{K}. \tag{71}$$

$$X_{pkij} \le h_{pk}, \quad \forall p\in\mathcal{P},\ i,j\in\mathcal{V},\ k\in\mathcal{K}. \tag{72}$$

$$\sum_{j\in\mathcal{V}} X_{pkD_k,j} = h_{pk}, \quad \forall p\in\mathcal{P},\ k\in\mathcal{K}. \tag{73}$$

$$\sum_{j\in\mathcal{V}\setminus i} X_{pkji} = \sum_{j\in\mathcal{V}\setminus i} X_{pkij} \le h_{pk}, \quad \forall p\in\mathcal{P},\ i\in\mathcal{J}\cup\bigcup_{a\in\mathcal{A}} D_a,\ k\in\mathcal{K}. \tag{74}$$

$$\sum_{j\in\mathcal{V}} X_{pkj,E_k} = h_{pk}, \quad \forall p\in\mathcal{P}\setminus\mathcal{P}_0,\ k\in\mathcal{K}. \tag{75}$$

$$\sum_{j\in\mathcal{V}} X_{pkj,E_H} = h_{pk}, \quad \forall p\in\mathcal{P}_0,\ k\in\mathcal{K}. \tag{76}$$

$$\sum_{p=p_{sjo}}^{p_{ejo}}\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}\setminus j} X_{pkij} = f_{jo}, \quad \forall j\in\mathcal{J},\ o\in\mathcal{P}_j. \tag{77}$$

$$B_{pj} - B_{pD_k} \le H_{pk}, \quad \forall p\in\mathcal{P},\ j\in E_H\cup E_k,\ k\in\mathcal{K}. \tag{78}$$

$$B_{pi} + s_i + t_{ij} \le B_{pj} + M\times(1-\sum_{k\in\mathcal{K}} X_{pkij}), \quad \forall p\in\mathcal{P},\ i,j\in\mathcal{V}. \tag{79}$$

$$\sum_{i\in\mathcal{J}\setminus j} X_{pkij} = \sum_{a\in\mathcal{A}} X_{pkj,D_a}, \quad \forall p\in\mathcal{P},\ j\in\mathcal{J}_\alpha,\ k\in\mathcal{K}. \tag{80}$$

$$L-\epsilon \le \sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{J}}\sum_{j\in\mathcal{J}\setminus i} X_{pkij} \le L+\epsilon, \quad \forall k\in\mathcal{K}. \tag{81}$$

$$\sum_{o\in\mathcal{P}_j}\sum_{p=p_{sjo}}^{p_{ejo}}\sum_{i\in\mathcal{J}\setminus j} X_{pkij} = \sum_{o\in\mathcal{P}_j} f_{jo}\times U_{kj}, \quad \forall j\in\mathcal{J},\ k\in\mathcal{K}. \tag{82}$$

$$\sum_{k\in\mathcal{K}} U_{kj} = 1, \quad \forall j\in\mathcal{J}. \tag{83}$$

$$B_{pj} = B_{pi}, \quad \forall p\in\mathcal{P},\ (i,\ j)\in\mathcal{J}_\gamma. \tag{84}$$

$$Y_{pij} + Y_{pji} \le \sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{V}} X_{pkli}, \quad \forall p\in\mathcal{P},\ (i,\ j)\in\mathcal{J}_\delta,\ j>i. \tag{85}$$

$$Y_{pij} + Y_{pji} \le \sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{V}} X_{pklj}, \quad \forall p\in\mathcal{P},\ (i,\ j)\in\mathcal{J}_\delta,\ j>i. \tag{86}$$

$$Y_{pij} + Y_{pji} \ge \sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{V}} X_{pkli} + \sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{V}} X_{pklj} - 1, \quad \forall p\in\mathcal{P},\ (i,\ j)\in\mathcal{J}_\delta,\ j>i. \tag{87}$$

$$B_{pi} + s_i + g_{ij} \leq B_{pj} + M \times (1 - Y_{pij}), \qquad \forall p \in \mathcal{P},\ (i,\ j) \in \mathcal{J}_{\delta},\ j > i. \tag{88}$$

$$B_{pj} + s_j + g_{ij} \leq B_{pi} + M \times (1 - Y_{pji}), \qquad \forall p \in \mathcal{P},\ (i,\ j) \in \mathcal{J}_{\delta},\ j > i. \tag{89}$$

$$X_{pkij}, Y_{pij}, U_{kj} \in \{0, 1\}, B_{pj} \geq 0, \qquad \forall p \in \mathcal{P},\ i,\ j \in \mathcal{V},\ k \in \mathcal{K}. \tag{90}$$

# OVERALL CONCLUSIONS

This research adopts the approach of mathematical modeling to solve three realistic and challenging vehicle routing problems, two in the context of public safety and law enforcement and the other in the context of home health care. The contributions of this dissertation include defining problems in a more systematic perspective, developing mixed integer linear models by the incorporation of realistic constraints, solving the models using either approximate or exact solution techniques and yielding good solution quality within short runtime, and providing managerial insights on how to improve routing efficiencies.

The first article applies a variant of the team orienteering problem with time windows. Our objective is to maximize the coverage time of state troopers. Due to the NP-hardness of our problem, we propose two heuristics, including a local search and a tabu search heuristics. We compare our solution with the ones using CPLEX, and our proposed heuristics can obtain objectives of good quality with fast speed. Using two heuristics, we conduct scenario analysis by varying the number of state troopers. The magnitudes of changes in the resulted objectives help us determine how many state troopers are needed per shift, per day, and per county.

In addition to evaluating the objectives, we evaluate two key performance measures: the percentage of hot spot covered and the percentage of coverage length. With the suggested number of state troopers, both performance measures are satisfactory, ranging between 85% and 100%, except that the percentage of hot spot covered is 63% for Jefferson county Monday evening shift and 76% for Tuscaloosa county Saturday evening shift. Under the single depot assumption, we find that the time windows of some HSs have already elapsed before the state troopers arrive even by traveling directly from the state trooper post. For these instances, even if the number of state troopers is unlimited, it is not possible to cover all of the HSs.

To improve the coverage, we relax the assumption of the single state trooper post in the first article. The second article incorporates multiple state trooper posts and multiple shifts. The existence of multiple state trooper posts makes it possible to select a state trooper post for each state trooper, which is a facility location problem. Hence, the extended problem is a combination of a multi-depot version of our first problem and the facility location problem, which is suitable to apply a decomposition-based heuristic. The decomposed master problem is solved by the solution approach of the first article with slight modifications to handle multiple depots. For the subproblems, we select the state trooper post with the smallest sum of the traveling cost and the depot's fixed cost. The overall problem is solved by iterating between the master problem and subproblems.

We use solutions obtained from CPLEX as the benchmark and demonstrate that our solution approach reduces the computational burden while maintaining good solution quality. Since the second article is an extension of the first one, we compare their results. We design 16 experiments with different parameter settings, and 15 of them find improvements in the objectives and the other one returns a tie between two solution approaches. When the HSs are uniformly distributed, on average the improvement in percentage amounts to as significant as 3.2%. When HSs are clustered distributed, on average DMD-MCPRP improves MCPRP by 9.6%. Moreover, the two predefined performance measures have satisfactory results. We also vary $D_{\text{limit}}$ from its initial value of 20 minutes and set it to a higher value of 240 minutes. We rerun the experiments, and DMD-MCPRP shows further improvements over MCPRP. Therefore, the need of this extension is justified.

The third article solves a home care scheduling problem with workload balance constraints. It explores a new vehicle routing problem which can be solved by column generation. Column generation is appropriate due to our large problem size. We test our solution approach with real home care data from February 3rd, 2012 to March 24th, 2012 with a rolling horizon, each of which is consisted of 7 days. We find the optimal solution for the LP relaxation of the problem and apply a simple rounding heuristic to satisfy the integrity constraints. The runtime is less than 1600 seconds. If the HCA schedules the caregivers as we suggest, the financial savings are significant in the first several weeks. Moreover, our solution reduces the the heaviest workload of each caregiver at the HCA from more than 40 patients to 25 patients per week. Therefore, our solution leads to much more equal workloads of caregivers, better employment satisfaction, and better services to patients. In addition, we compare the solutions when centrifuges are available with the ones when they are not available. The results reveal that the savings are not sufficient to cover the investment cost, so the HCA should not purchase centrifuges.

Our solution approach is capable of solving real size problems with short runtime. We believe that our modeling approach is generic so that it can be easily adapted to other home care problems.

The research work in this dissertation can be extended in a number of ways:

- In the first topic, we will relax the assumption of the constant travel speed. Specifically, we will assume that it follows a certain probability distribution. This will convert the problem from a deterministic problem to a stochastic problem, which should be solved via stochastic optimization techniques.

- Simulated annealing is used to solve the second article. We can improve its solution quality by adopting some other contemporary metaheuristics with a learning mechanism, such as the ant colony optimization, greedy randomized adaptive search procedures, etc.

- In the second topic, we will relax the assumption that the HCA only approves caregivers' absences at least one day ahead of time, which we will assume to happen with a certain probability. Each request is assigned to both a primary caregiver and a backup caregiver. As a result, when the primary caregiver is not available, the backup caregiver is assigned to fulfill her duty. The joint probability for both the primary and backup caregivers to be absent on the same day is negligible. Overall, one request is treated by two caregivers at most. This extension turns the problem into a stochastic problem, and the objective is to minimize the expected travel cost.

- Apply some metaheuristics on the neighborhood search algorithms of the third article to solve the subproblem faster.

- Finally, the solution quality of the third article can be improved if we combine the column generation with the branch-and-bound technique, on the condition that the runtime at each node is shortened.

REFERENCES

Burnson, P. 2011. 22nd annual state of logistics report: A bunpy ride. *Council of Supply Chain Management Professionals*.

Clarke, G., J. W. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* **12**(4) 568–581.

Dantzig, G. B., J. H. Ramser. 1959. The truck dispatching problem. *Management Science* **6**.(1) 80–91.

Desrochers, M., J. K. Lenstra, M. W. P. Savelsbergh. 1990. A classification scheme for vehicle routing and scheduling problems. *European Journal of Operational Research* **46**(3) 322–332.

Golden, B. L., S. Raghavan, E. A. Wasil. 2008. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York.

Laporte, G. 1992. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* **59**(3) 345–358.

Laporte, G. 2009. Fifty years of vehicle routing. *Transportation Science* **43**(4) 408–416.

Laporte, G., I. H. Osman. 1995. Routing problems: A bibliography. *Annals of Operations Research* **61**(1) 227–262.

Toth, P., D. Vigo. 2002. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia.