Transportation Research Part E 48 (2012) 215–232

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre

Analysis of an integrated maximum covering and patrol routing problem

Burcu B. Keskin [a,*], Shirley (Rong) Li [a], Dana Steil [b], Sarah Spiller [c]

[a] Department of Information Systems, Statistics, and Management Science, University of Alabama, Tuscaloosa, AL 35487-0226, United States
[b] Department of Computer Science, University of Alabama, Tuscaloosa, AL 35487-0290, United States
[c] Department of Economics, Finance, and Legal Studies, University of Alabama, Tuscaloosa, AL 35487-0224, United States

ARTICLE INFO

Article history:
Received 25 July 2010
Received in revised form 10 June 2011
Accepted 16 June 2011

Keywords:
Maximum covering
Patrol routing
Time window
Public service
Local search
Tabu search

ABSTRACT

In this paper, we address the problem of determining the patrol routes of state troopers for maximum coverage of highway spots with high frequencies of crashes (hot spots). We develop a specific mixed integer linear programming model for this problem under time feasibility and budget limitation. We solve this model using local and tabu-search based heuristics. Via extensive computational experiments using randomly generated data, we test the validity of our solution approaches. Furthermore, using real data from the state of Alabama, we provide recommendations for (i) critical levels of coverage; (ii) factors influencing the service measures; and (iii) dynamic changes in routes.

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 (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 US roadways is still noteworthy. The NHTSA estimates this cost as $230.6 billion per year (nearly 2.3% 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) (Feillet

* Corresponding author. Tel.: +1 205 348 8442.
E-mail address: bkeskin@cba.ua.edu (B.B. Keskin).

1366-5545/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.tre.2011.07.005

et al., 2005; Tsiligirides, 1984), 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 1 min or as large as the length of the time window (up to 270 min).

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 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 2, we present the literature review. In Section 3, we present the general mathematical model, including necessary assumptions and notation. In Section 4, we present the analysis of the problem and the solution approaches based on the characteristics of the problem. In Section 5, we discuss the computational results based on randomly generated data and real data. Finally, in Section 6, we provide our conclusions, recommendations, and future work.

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; McCullagh, 2006; Miranda-Moreno et al., 2007; 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.

2.1. State trooper/police 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 (Birge and Pollock, 1989; Chaiken and Dormont, 1978; Green, 1984; Larson, 1973). Other approaches for the patrol routing problem include mathematical modeling (Curtin et al., 2007; Mitchell, 1972; Calvo and Cordone, 2003), heuristic solutions (Lauri and Koukam, 2008; Reis et al., 2006; Calvo and Cordone, 2003), graph theory (Chawathe, 2007; Duchenne et al., 2005; Duchenne et al., 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. 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

[1] CPLEX is a trademark of IBM.

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.

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., 1996). 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., 1996), 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, 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; Schilde et al., 2009; Tang and Miller-Hooks, 2005), path relinking (Schilde et al., 2009; Souffriau et al., 2010), ant colony optimization (Ke et al., 2008; Liang et al., 2002; 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).

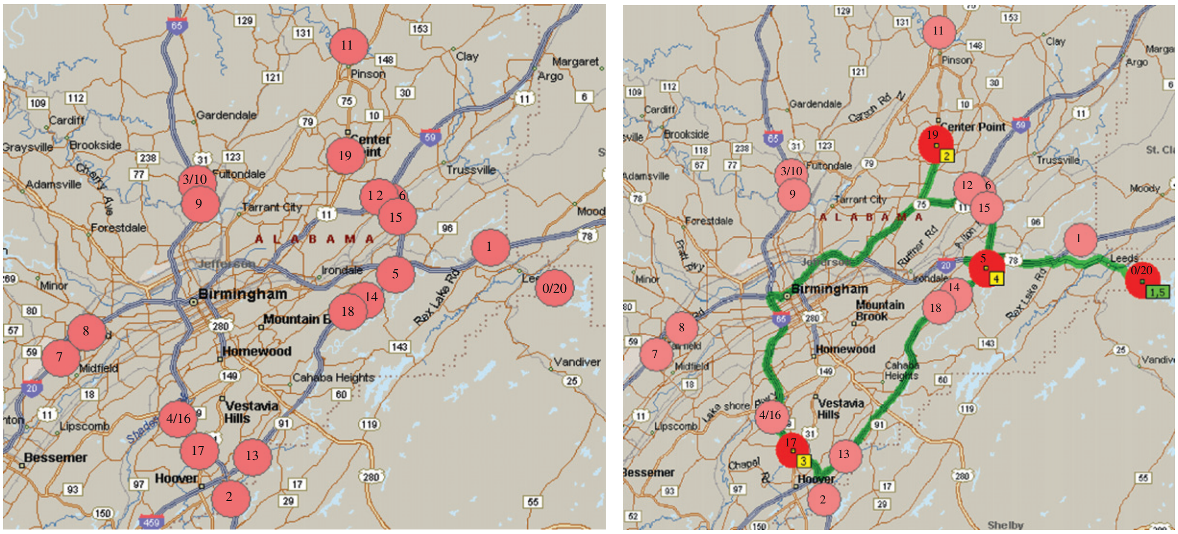
As our problem bears similarities to the TOPTW, we discuss the TOPTW literature in more detail. The exact branch-and-price algorithm proposed by Boussier et al. (2007) is generic enough to handle different kinds of OP, including the TOPTW. The different branching rules and acceleration techniques introduced 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 to 20 routes and 48 to 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.

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 $e_1 \leqslant e_2 \leqslant \cdots \leqslant e_n$. 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 Fig. 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 7:00–8:30 am and 11:00 am–12:30 pm, respectively. In Fig. 1b, 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.

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



(a) Hot spots (b) An optimal route

Fig. 1. A representative example.

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) $s_{ik} \geqslant 0$, the starting time of service for state trooper car $k \in \mathcal{K}$ at vertex $i \in \mathcal{V}$. (iii) $f_{ik} \geqslant 0$, 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/h. 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.

3.1. 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 $s_{jk} \geqslant f_{ik} + t_{ij}$. To ensure schedule feasibility, we need

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

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

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

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

Other schedule feasibility constraints include time window restrictions:

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

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

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

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.

3.2. Route structuring constraints

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

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

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

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

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.

3.3. Constraints related to visiting hot spots

It is possible to have multiple cars visiting the same hot spot as in Fig. 2b and c. Therefore, we need to account for any potential double counting if there is overlap during the visits of multiple cars, as in Fig. 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 $k, g \in \mathcal{K},\ k \neq g$:

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

By definition of y_{ik},

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

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

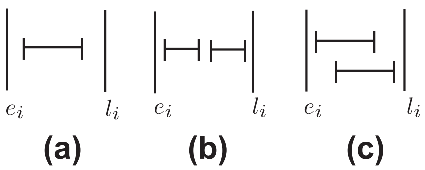


Fig. 2. Multiple state troopers at hot spot $i \in \mathcal{N}$.

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}:

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

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

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

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 $f_{ik} \leqslant s_{ig}$ or $s_{ik} \geqslant f_{ig}$, where $k,g \in \mathcal{K}$ and $k \neq g$:

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

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

where M is a large constant.

3.4. Integrality and non-negativity constraints

Finally, we state continuous and binary variables:

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

3.5. 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.

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 $i \in \mathcal{N}$*

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

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/h 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.

4.1. Construction algorithm

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

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 min, which is reasonable given that for the instances we tested $T = 480$ min. If the travel time from a currently visited hot spot to the next one exceeds 100 min, 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: $Route_k \leftarrow Route_k \cup \{0\}$.
4: $i^* \leftarrow \arg\max_{i \in \mathcal{U}}\{l_i - \max(e_i, t_{0i}) : i \leqslant H_{limit}, t_{0i} \leqslant T_{limit}, t_{0i} \leqslant l_i\}$.
5: $s_{i^*,k} \leftarrow \max\{e_{i^*}, t_{0,i^*}\}$ 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: $i \leftarrow Route_k.lastHotSpot$.
9: **for** $\forall j \in \mathcal{U}$ such that $i < j \leqslant (i + H_{limit}), t_{ij} \leqslant T_{limit}$, and $l_i + t_{ij} < l_j$ **do**
10: **if** $l_j + t_{j,n+1} < T$ **then**
11: $i^* \leftarrow \arg\max_{j \in \mathcal{U}}\{l_j - \max(e_j, l_i + t_{ij})\}$.
12: $s_{i^*,k} \leftarrow \max\{e_{i^*}, l_i + t_{i,i^*}\}$ and $f_{i^*,k} \leftarrow l_{i^*}$. $Route_k \leftarrow Route_k \cup \{i^*\}$.
13: **if** $e_{i^*} < l_i + t_{i,i^*}$ **then**
14: $l_{i^*} \leftarrow l_i + t_{i,i^*}$.
15: **else**
16: $\mathcal{U} \leftarrow \mathcal{U} \setminus \{i^*\}$.
17: **end if**
18: **else**
19: **if** $l_i + t_{ij} < T - t_{j,n+1}$ **then**
20: $i^* \leftarrow \arg\max_{j \in \mathcal{U}}\{T - t_{j,n+1} - \max(e_j, l_i + t_{ij})\}$;
21: $s_{i^*,k} \leftarrow \max\{e_{i^*}, l_i + t_{i,i^*}\}$ and $f_{i^*,k} \leftarrow T - t_{i^*,n+1}$. $Route_k \leftarrow Route_k \cup \{i^*\}$.
22: Repeat Steps 13 to 17.
23: **end if**
24: **end if**
25: **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 13–17 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 13–17 are repeated. In this way, initial $|\mathcal{K}|$ routes are created in parallel.

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:

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

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.

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 (Braysy and Gendreau, 2005a; 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.

4.2.1. Relocate operator

In Fig. 3a, 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 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

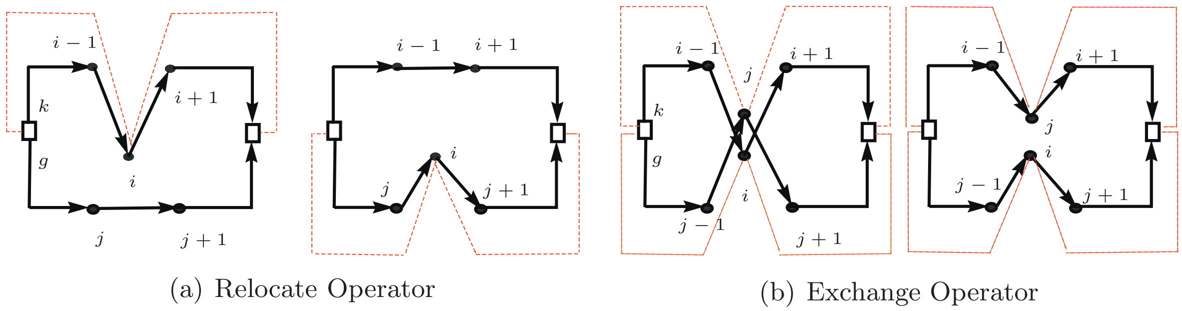


Fig. 3. Neighborhood search operators.

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 $\Delta \leqslant 0$, 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 Eq. (16). If $\delta_1 > 0$, the insertion before $i+1$ is accepted and updated benefit $\widehat{\Delta}$ is set as $\Delta + \delta_1$. Otherwise, if $\delta_2 > 0$, the insertion before i is accepted and $\widehat{\Delta}$ is set as $\Delta + \delta_2$. Finally, if $\delta_3 > 0$, $\widehat{\Delta}$ is set as $\Delta + \delta_3$. If none of the insertions are favorable, i.e. $\delta_a < 0$ for $a = 1,2,3$, the $\widehat{\Delta}$ is the same as Δ. Among all of the positive $\widehat{\Delta}$ 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 $\widehat{\Delta}$. That is, we use the Global Best (GB) acceptance rule.

4.2.2. Exchange operator

In Fig. 3b, we present the exchange operator where two hot spots i and j swap routes simultaneously. As in Fig. 3a, 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

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

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 δ_1, δ_2, δ_3, and δ_4, respectively, as in Eq. (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 $\widehat{\Delta}$ is updated as $\Delta + \delta_a$. If none of the insertions return a benefit, then $\widehat{\Delta}$ is just set to Δ. Similar to relocate operator, the exchange operator is implemented using the GB criteria. The exchange (and potential insertion) associated with the largest $\widehat{\Delta}$ 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.

4.2.3. Local search

After introducing all of the neighborhood search components, Fig. 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.

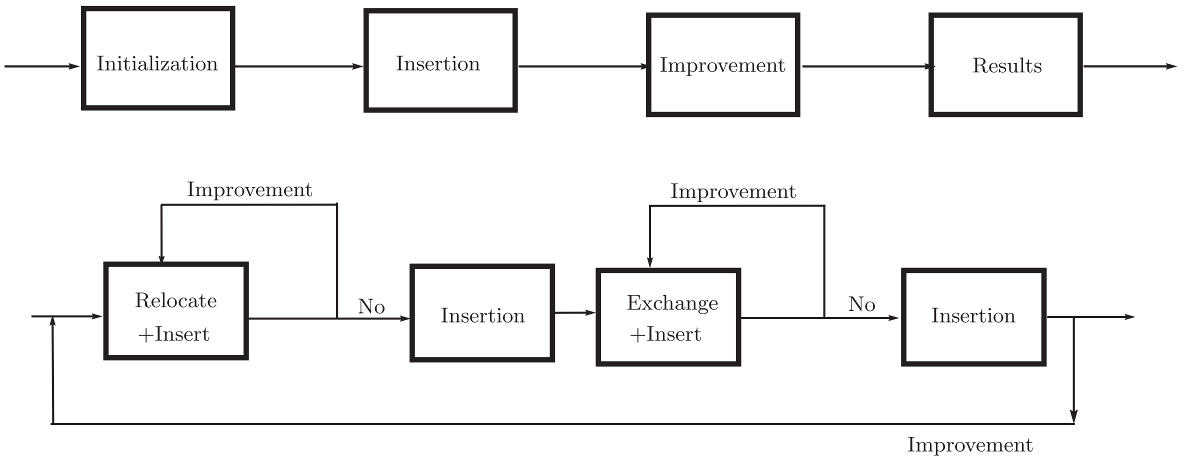


Fig. 4. Local search and improvement flow charts.

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 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.

5. Computational experiments

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 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 3.66 GHz Xeon processors and 8 GB of memory.

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 **Gap** = $(Objective - LB)/LB * 100$. 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".

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

Table 1
Performances of LS and TS for random data.

Data set	No. cars	No. instances	CPLEX no. opt.	LS			TS		
				Avg.	Max.	No. best	Avg.	Max.	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

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 s 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 h 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.

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 7:00 am to 3:00 pm, afternoon shift from 3:00 pm to 11:00 pm, and evening shift from 11:00 pm to 7:00 am. As the other weekdays (Tuesday–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.

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.

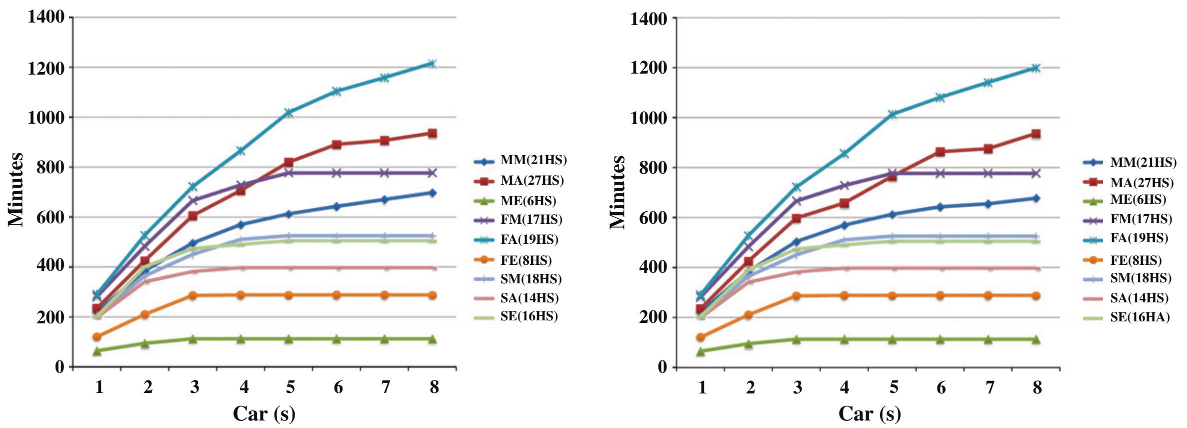
On the other extreme, there is a positive gap of 19.5% for Mobile during Saturday afternoon shift with four state trooper cars. This is attributed to the poor performance of CPLEX. However, this is not due to our formulation or the gap. More specifically, for this instance as well as the instances marked in bold, 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

Table 2
Performance 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

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.



(a) Local Search (b) Tabu Search

Fig. 5. The coverage with LS and TS due to different state trooper cars in Jefferson County.

5.2. Managerial insights

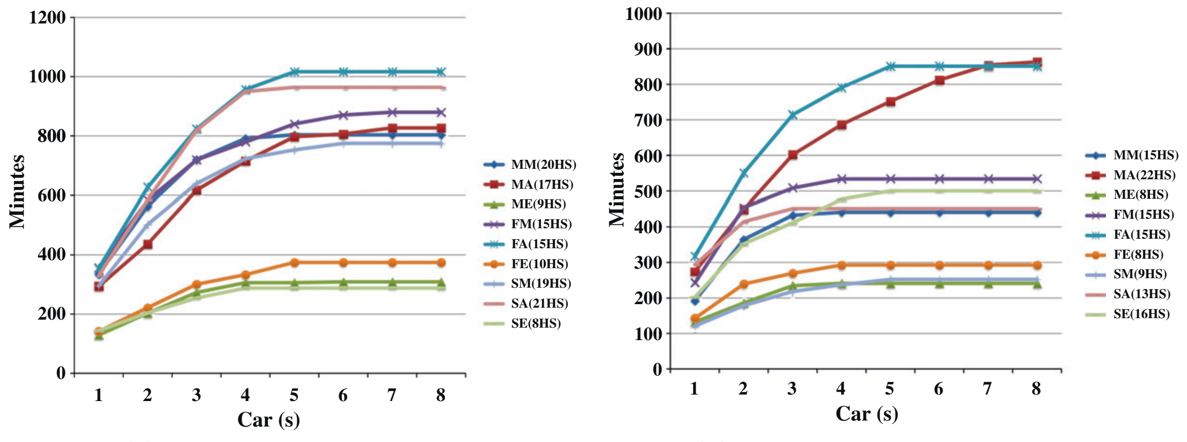
In this section, we provide managerial insights for decision makers based on our solutions with real data. In Figs. 5 and 6, we plot the objective value of *MCPRP* returned by LS and TS with respect to different state trooper cars, respectively. From the plotted charts, we can determine how many state 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 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 Figs. 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 Fig. 5, the objective of Monday afternoon is not concave, since they are returned by our heuristics.

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.



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

Fig. 6. The coverage with LS in the city of Mobile and Tuscaloosa County.

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

Table 4
LS performance 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

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.

Acknowledgements

The authors thank the University Transportation Center for Alabama for providing support for this research under the Grant UTCA-09103. The authors also thank the Center for Advanced Public Safety at the University of Alabama (http://care.cs.ua.edu/) for providing the data for this research. The authors acknowledge the Computer-based Honors program of the University of Alabama for providing support for Ms. Sarah Spiller during her undergraduate research project. Lastly, the authors thank the two anonymous referees and the editor for their constructive feedback.

Appendix A

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 α_i is an arbitrary small number, say 1 min, such that the stop at hot spot i can only collect α_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.
Decision variables	
X_j	1, if route j is selected as a part of patrolling plan, 0, otherwise
Y_i	1, if hot spot i is covered, 0, otherwise

The mathematical formulation is presented as

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

$$\text{s.t.} \quad \sum_{j\in N_i} X_j \geqslant Y_i, \quad \forall i \in \mathcal{I} \tag{18}$$

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

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

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 $p_1 \geqslant 1$ 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 $n \times (n - p_1) \times (n - p_2) \times \cdots \times (n - p_k)$, where $p_k > p_{k-1} > \cdots > p_2 > p_1$ 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 $\upsilon(S^*)$. 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 $f_{ik} + t_{i,n+1} \leqslant 480$. 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', $\upsilon(S')$, is larger than $\upsilon(S^*)$, 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 $\upsilon(S^*) - \upsilon(S') = f_{ik} - l_i - s_{i+1,k} + s'_{i+1,k}$, where $s'_{i+1,k}$ is the start time at hot spot $i+1$ on route k in solution S'. Now,

$$s'_{i+1,k} - s_{i+1,k} = \max(l_i + t_{i,i+1}, e_{i+1}) - \max(f_{ik} + t_{i,i+1}, e_{i+1}) = \begin{cases} l_i - f_{ik} & \text{if } e_{i+1} \leqslant f_{ik} + t_{i,i+1} \\ l_i + t_{i,i+1} - e_{i+1} & \text{if } f_{ik} + t_{i,i+1} < e_{i+1} \leqslant l_i + t_{i,i+1} \\ 0 & \text{if } e_{i+1} > l_i + t_{i,i+1} \end{cases}$$

Note that in all cases, $s'_{i+1,k} - s_{i+1,k} \leqslant l_i - f_{ik}$. Therefore, $\upsilon(S^*) - \upsilon(S') < 0$, 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, $t_{i,i+2} \leqslant t_{i,i+1} + t_{i+1,i+2}$. Then, for route k, the difference in the objective functions $\upsilon(S^*)$ and $\upsilon(S')$ 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 $v(S') \geqslant v(S^*)$ 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 $v(S'_b) - v(S^*) \geqslant 0$. We need to prove that if $b + 1$ hot spots are eliminated, $v(S'_{b+1}) \geqslant v(S_{b+1})$ holds. From the triangular inequality, we know $t_{i,i+b+1} \leqslant t_{i,i+b+1} + t_{i+b+1,i+b+2} \leqslant t_{i,i+1} + t_{i+1,i+2} + \cdots + t_{i+b-1,i+b} + t_{i+b,i+b+1} + t_{i+b+1,i+b+2}$. 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,

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

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, $v(S'_{b+1}) \geqslant v(S^*)$. Hence, the modified solution is as good as S^*.

This concludes the proof. □

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

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

Blincoe, L., Seay, A., Zaloshnja, E., Miller, T., Romano, E., Luchter, S., Spicer, R., 2002. The Economic Impact of Motor Vehicle Crashes, 2000. Crash Statistics DOT HS 809 446, NHTSA.

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

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

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

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

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

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

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

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

Chen, H.C., Quddus, M.A., 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., Washington, S.P., 2005. Experimental evaluation of hotspot identification methods. Accident Analysis and Prevention 37 (5), 870–881.

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

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

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

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

Feillet, D., Dejax, P., Gendreau, M., 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., Bailey, T.C., Diggle, P.J., Rowlingson, B.S., 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., Levy, L., Vohra, R., 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., Archetti, C., Feng, Z., 2008. Ants can solve the team orienteering problem. Computers and Industrial Engineering 54, 648–665.

Laporte, G., Martello, S., 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., Koukam, A. 2008. A two-step evolutionary and ACO approach for solving the multi-agent patrolling problem. In: IEEE Congress on Evolutionary Computation, 2008. CEC 2008 (IEEE World Congress on Computational Intelligence), pp. 861–868.

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

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

Marianov, V., ReVelle, C., 1995. Siting emergency services. In: Drezner, Z. (Ed.), Facility Location. Springer, Berlin, pp. 199–223.

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

Miranda-Moreno, L.F., Labbe, A., Fu, L., 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., Gambardella, L., 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., Melo, A., Coelho, A.L.V., Furtado, V., 2006. Towards optimal police patrol routes with genetic algorithms. Lecture Notes in Computer Science 3975, 485–491.
Righini, G., Salani, M., 2006. Dynamic programming for the orienteering problem with time windows. Technical Report 91, Universita degli Studi, Milano, Crema, Italy.
Righini, G., Salani, M., 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., Ramalho, G., Corruble, V., Ratitch, B., 2004. Multi-agent patrolling with reinforcement learning. In: AAMAS-2004 – Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems, vol. 3, pp. 1122–1129.
Schilde, M., Doerner, K.F., Hartl, R.F., Kiechle, G., 2009. Metaheuristics for the bi-objective orienteering problem. Swarm Intelligence 3 (3), 179–201.
Souffriau, W., Vansteenwegen, P., Berghe, G.V., Oudheusden, D.V., 2010. A path relinking approach for the team orienteering problem. Computers and Operations Research.
Steil, D., Parrish, A., 2009. Hit: a GIS-based hotspot identification taxonomy. International Journal of Computers and their Applications 16 (2), 1–10.
Tang, H., Miller-Hooks, E., 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., Romauch, M., Doerner, K.F., Hartl, R.F., 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., Souffriau, W., Berghe, G.V., Oudheusden, D.V., 2009. Iterated local search for the team orienteering problem with time windows. Computers and Operations Research 36 (12), 3281–3290.
Vansteenwegen, P., Souffriau, W., Oudheusden, D.V., 2010. The orienteering problem: a survey. European Journal of Operational Research.