Journal of the Operational Research Society (2014) 65, 1711–1725

www.palgrave-journals.com/jors/

Bi-criteria dynamic location-routing problem for patrol coverage

Shirley (Rong) Li and Burcu B Keskin*

The University of Alabama, Tuscaloosa, USA

In this paper, we address the problem of dynamic patrol routing for state troopers for effective coverage of highways. Specifically, a number of state troopers start their routes at temporary stations (TS), patrol critical locations with high crash frequencies, and end their shifts at other (or the same) TS so the starting points for the next period are also optimized. We determine the number of state troopers, their assigned routes, and the locations of the TS where they start and end their routes. The TS are selected from a given set of potential locations. The problem, therefore, is a multi-period dynamic location-routing problem in the context of public service. Our objective is to maximize the critical location coverage benefit while minimizing the costs of TS selections, vehicle utilizations, and routing/travel. The multi-objective nature of the problem is handled using an ε-constraint approach. We formulate the problem as a mixed integer linear programming model and solve it using both off-the-shelf optimization software and a custom-built, efficient heuristic algorithm. The heuristic, utilizing the hierarchical structure of the problem, is built on the decomposition of location and routing problems. By allowing routing to start from multiple locations, our model improves the coverage by as much as 12% compared with the single-depot coverage model.

Journal of the Operational Research Society (2014) **65**(11), 1711–1725. doi:10.1057/jors.2013.116

Published online 6 November 2013

Keywords: bi-criteria; multi-period; location-routing problem; heuristics; public service; practice of OR

1. Introduction

Traffic incidents not only greatly impact individuals, but also affect the general population. Depending on the incident severity, incidents are likely to cause both private and public property damage and possibly cause injury and fatalities. Due to both economic and humanitarian importance, maintaining roadway travel safety has become a research interest for government officials, industry, and researchers. The National Highway Traffic Safety Administration estimates the cost of improving safety through various law enforcement activities to be as high as US$230.6 billion annually, nearly 2.3% of the nation's gross domestic product (Blincoe *et al*, 2002). Recently, a number of researchers (Steil and Parrish, 2009; Lou *et al*, 2011; Keskin *et al*, 2012; Willemse and Joubert, 2012) have studied the effectiveness of law enforcement plans, including how to improve patrolling plans. One way of improving patrolling efficiency is to focus on patrolling critical locations with high crash frequencies.

Given historical crash data, a 'hot spot' (HS) is defined as a certain stretch of highway with a high frequency of crashes of different severity over a given time period. In this problem, we are interested in finding the right start and stop locations (temporary stations (TS)) 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 state troopers during the *hot times* of the HSs while minimizing the costs associated with the number of state troopers dispatched, travelling from one HS to another, and potential fees for TS. Therefore, we tackle a bi-criteria (benefit maximization and cost minimization) optimization problem.

Specifically, we assume that at the beginning of a shift state troopers start their patrol at TS whose locations need to be determined from a list of potential stations. In reality, the state troopers take their cars to their homes and start and end their shift at parking lots of public or retail facilities that are located within the vicinity of their homes or at the state trooper posts. Hence, the list of such facilities comprises the list of potential TS. Starting at these selected TS, the state troopers travel from one HS to another during their shift and stop at HSs during the effective coverage time, that is, during the time interval that a particular HS is critical. At the end of the shift, the state troopers go to other TS 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 TS need to be determined as well. A state trooper can only relocate to another potential TS that is within a certain vicinity of the starting temporary location. Depending on how *vicinity* is measured, a state trooper may have abundant or a limited

**Correspondence: Burcu B Keskin, Department of Information Systems, Statistics, and Management Science, The University of Alabama, 355 Alston Hall, Tuscaloosa, AL, 35487–0226, USA.*
E-mail: bkeskin@cba.ua.edu

This research has been supported by the University Transportation Center for Alabama Grants UTCA-09104 and UTCA-11113.

number of choices for the relocation of the TS. With these characteristics, this problem is similar to a *multi-depot (multiple TS)*, *dynamic location (changing locations) and location-routing problem* (LRP). At the same time, since the presence of the state troopers at the defined HSs, that is, the service time at a HS, can be viewed as the 'variable profit', the problem resembles the *team orienteering problem with time windows* (TOPTW).

We note that our paper is closely related to the work by Keskin *et al* (2012) that focuses on patrol coverage of HSs. Considering a single station, Keskin *et al* (2012) propose 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, even with an unlimited 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, the effective coverage will have already lapsed. Our work extends their paper in three directions:

1. We consider multiple TS whose locations need to be determined as opposed to a single depot. This way, more, if not all, HSs are covered that are located out of the accessibility range with just one station.
2. Our model spans multiple periods (shifts) as the locations of the HSs and TS dynamically change and TS locations tie the multiple periods together.
3. In addition to 'coverage benefit' maximization, we also consider the minimization of total system costs (cost of selection of troopers, travel costs, and TS location costs). With the addition of TS, 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 budget allocation and HS coverage. For instance, if fewer state troopers are dispatched or fewer TS are available, 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 TS are opened, there may not be enough monetary resources to pay for patrolling costs.

Since the MCPRP, which arises as a subproblem of our problem, and the 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 neighbourhood searches embedded within a simulated annealing (SA) framework. 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. In addition, we provide additional service measures including the percentage of number of HSs covered and percentage of coverage length based on the outcome of the heuristics, as defined in Keskin *et al* (2012). These service measures provide additional insights into the solutions.

The remainder of this paper is structured as follows. In Section 2, we present the literature review. In Section 3, the details of the general mathematical model are discussed, including necessary assumptions and notations. Next, 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 the heuristics and their implications. Finally, in Section 6, we summarize our results and offer recommendations for effective implementation.

2. Literature review

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

2.1. TOPTW

The OP is first introduced by Tsiligirides (1984) for the orienteering competition. The goal is to identify 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 where a fixed number of paths is considered, the orienteering problem with time windows (OPTW), and the TOPTW.

Boussier *et al* (2007), Montemanni and Gambardella (2009), and Vansteenwegen *et al* (2009) are the only researchers known to have solved the TOPTW. The exact branch-and-price algorithm proposed by Boussier *et al* (2007) is generic enough to handle different kinds of OP, including the TOPTW. Montemanni and Gambardella (2009) develop 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 the best of our knowledge, only Tricoire *et al* (2010) work on a multi-period OPTW problem. They design a variable neighbourhood search-based metaheuristic. However, they specify a fixed starting depot and a fixed stopping depot for each period for only one car, whereas in our case the starting and ending locations are decision variables for multiple state troopers.

2.2. LRP

Since Salhi and Rand (1989) show that LRP consistently produces better solutions than solving sub-problems of facility location and vehicle routing sequentially, LRP has received increased attention from researchers. Laporte (1988) summarizes two-index or three-index vehicle flow formulations for static, deterministic LRP. For more information, please refer to the reviews by Balakrishnan *et al* (1987), Min *et al* (1998), and Nagy and Salhi (2007).

Both exact algorithms and heuristics are designed to solve LRP, but exact algorithms (see Labbé *et al* (2004) and Laporte *et al* (1986)) are still limited to small- to medium-sized 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); Wu *et al* (2002)), and hierarchical heuristics (Albareda-Sambola *et al* (2007); Melechovský *et al* (2005); Nagy and Salhi (1996)). Among these four categories, the last two are preferred as sequential and clustering heuristics fail to utilize feedback between location and routing sub-problems. Especially when there is hierarchy involved, hierarchical heuristics are shown to be more effective. A hierarchical heuristic divides LRP into a master problem location and its subordinate routing problem. We follow this logic in the development of our heuristics.

We note two important differences between our work and the literature. First, our problem has a time window limitation that has not yet been addressed in the LRP literature to the best of our knowledge. 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 Vehicle Routing Problem with Time Windows (VRPTW) literature instead of LRP since there are no location decisions. Second, instead of locating long-term depots, we locate TS while optimizing routing schedules. For our problem, both location and routing are short-term decisions, avoiding the common criticism that LRP has conflicting planning horizons in the short and long run.

3. General model

3.1. Problem definition

As discussed earlier, it is assumed that at the beginning of each shift, state trooper cars are dispatched from TS, where the potential locations are given as $i \in \mathcal{I} = \{1, 2, \ldots, |\mathcal{I}|\}$, which also includes state trooper posts. State trooper posts, similar to a police station, are physical buildings where the troopers report for desk duty when they are not on patrol duty. The selection of TS i may incur a fixed cost $F_i \geqslant 0$, especially if, for instance, there is a parking fee for using i. $\mathcal{K} = \{1, 2, \ldots, |\mathcal{K}|\}$ is the set of the available state troopers, and each trooper on patrol 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 $\mathcal{D} = \{1, 2, \ldots, |\mathcal{D}|$ be the set of days, where 1 represents the first day and so on. As a simplification, pairs of $p \in \mathcal{P}$ and $d \in \mathcal{D}$ can be represented by a single-period index $t \in \mathcal{T} = \{1, \ldots, \mathcal{T}\}$, where $T = |\mathcal{P}| \times |\mathcal{D}|$.

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, $j = 1, \ldots, n_t$. In a period $t \in \mathcal{T}$, HS j has three attributes: (i) location on the mile-posted road network; (ii) 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 'hot' time window; and (iii) weight w_j^t representing severity level. By definition, $e_j^t \leqslant l_j^t$. Furthermore, we assume without loss of generality that $\mathcal{N}^t$ is an ordered set according to e_j^t such that $e_1^t \leqslant e_2^t \leqslant \ldots \leqslant e_n^t$. Note that a location can be listed as two different HSs i and j, where $e_i^t < e_i^t$ if it becomes 'hot' twice within the same period.

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

- $\Delta^+(i) = \{j \in \mathcal{V}^t, t \in \mathcal{T} : (i, j) \in \mathcal{A}^t, e_i^t + d_{ij}^t \leqslant l_j^t\}$ as the set of vertices that are directly reachable from $i \in \mathcal{V}$ within the time window, and
- $\Delta^-(i) = \{j \in \mathcal{V}^t, t \in \mathcal{T} : (j, i) \in \mathcal{A}^t, e_j^t + d_{ij}^t \leqslant l_i^t\}$ as the set of vertices from which i is directly reachable.

The additional assumptions of the model are as follows:

1. There is a fixed mileage cost for patrolling, \$$c$ per mile.
2. There is no capacity limit at TS, that is, 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 a 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. If a state trooper does not come back to the same TS, the new TS can only be D_{limit} miles from the TS of the current period.
6. State trooper cars travel at a constant speed of φ miles/hour, and thus d_{ij}^t miles can be converted into $60d_{ij}^t/\varphi$ min. This way, distance and time can be easily translated to one other.
7. State troopers can choose whether or not to visit an HS, as well as the time to begin and end the coverage. If an HS is chosen by a state trooper, it cannot be visited by others.
8. A state trooper does not need to stay at an HS for the whole duration of the time window.

Our goal is to optimize the dynamic selection of TS utilized each period, allocate state troopers to TS, and route state troopers to HSs simultaneously. Figure 1 shows an example with 5 potential TS, 3 available state troopers, 2 periods, and 16 HSs per period. At the beginning, these 3 cars are parked at TS 2, 3, and 5. 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 three troopers are utilized; when $t = 2$, only two troopers are utilized due to budget limitations. That is, the cost minimization outplays the benefit

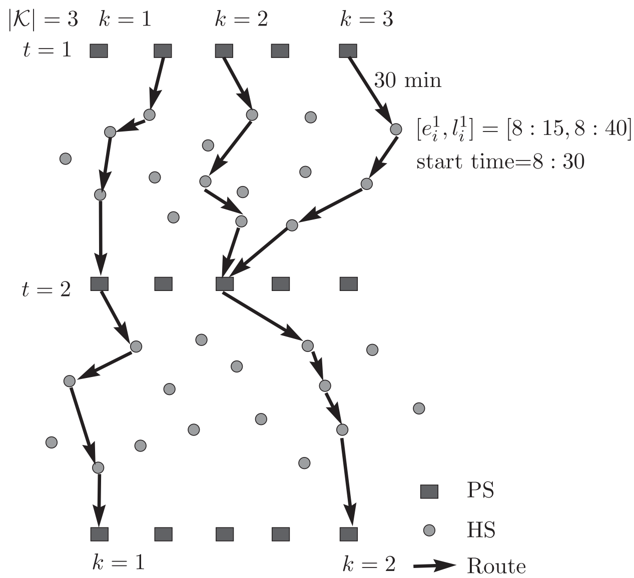

Figure 1 A representative example.

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.

3.1.1. Decision variables. We define six sets of decision variables: (i) $x^t_{ijk}=1$, if state trooper car $k = \mathcal{K}$ travels from i to j, $(i, j) \in \mathcal{A}^t$ during $t \in \mathcal{T}$, and 0, otherwise; (ii) $s^t_{ik} \geqslant 0$, the starting time of service for state trooper car $k \in \mathcal{K}$ at $i \in \mathcal{V}^t$ during $t \in \mathcal{T}$; (iii) $f^t_{ik} \geqslant 0$, the time state trooper car $k \in \mathcal{K}$ leaves $i \in \mathcal{V}^t$ during $t \in \mathcal{T}$, that is, the end of service; (iv) $y^t_{ik}=1$, if state trooper k serves $i \in \mathcal{V}^t$ during $t \in \mathcal{T}$, 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 $t \in \mathcal{T}$, $i, j \in \mathcal{I}$, 0, otherwise; (vi) $z^t_i=1$, if TS $i \in \mathcal{I}$ is open in $t \in \mathcal{T}$, 0, otherwise.

3.1.2. Objective. We have a multi-objective optimization problem, including cost (trooper salary 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 period (shift).

Then, our objective is:

$$\min_{x,z} \left(v \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{N}^t} \sum_{k\in\mathcal{K}} x^t_{ijk} + c \sum_{t\in\mathcal{T}} \sum_{(i,j)\in\mathcal{A}^t} \sum_{k\in\mathcal{K}} d^t_{ij} x^t_{ijk} + \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} F_i z^t_i \right) \tag{1}$$

$$\max_{f,s} \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{N}} \sum_{k\in\mathcal{K}} (f^t_{jk} - s^t_{jk}) w^t_j \tag{2}$$

For multi-objective optimization problems, it is very common that objectives may not be commensurate with each other. Similarly, for our problem, the coverage benefit is measured in minutes whereas the total cost is measured in dollars. Facing a similar dilemma, the vast majority of researchers use either a weighted sum of the objectives or the ε-constraint approach. The first group of researchers (Alumur and Kara, 2007; Caballero *et al*, 2007; Alçada-Almeida *et al* 2009) transformed conflicting objectives into a weighted sum by attaching each objective to a coefficient. However, due to the arbitrary choices of coefficients, we adopt the other commonly used method: the ε-constraint approach (Chankong and Haimes, 1983; Miettinen, 1999; Laumanns *et al*, 2005, 2006; Bérubé *et al*, 2009; Mavrotas, 2009). This approach considers the single most important objective and puts all of the other objectives into the formulation as constraints. Thereafter, our problem is transformed into a benefit maximization problem by setting an upper limit on the budget, say $\mathcal{B}$, that is, $\left(v \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{N}^t} \sum_{k\in\mathcal{K}} x^t_{ijk} + c \sum_{t\in\mathcal{T}} \sum_{(i,j)\in\mathcal{A}^t} \sum_{k\in\mathcal{K}} d^t_{ij} x^t_{ijk} + + \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} F_i z^t_i \right) \leqslant \mathcal{B}$. In our computational experiments, we test different levels of $\mathcal{B}$ to demonstrate the effect of costs and available budgets on the patrol routes.

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 Constraint (4), and integrality and non-negativity Constraints (6a)–(6b). In Constraint (1a), $M^t_{ij} = \max\{l^t_i + d^t_{ij} - e^t_j, 0\} \geqslant 0$, and in Constraint (3d), D_{limit} is a constant.

Schedule feasibility

$$f^t_{ik} + d^t_{ij} - s^t_{jk} \leqslant (1 - x^t_{ijk}) M^t_{ij}, \quad \forall t \in \mathcal{T}, (i,j) \in \mathcal{A}^t, k \in \mathcal{K}. \tag{1a}$$

$$e^t_i \times \sum_{j\in\Delta^+(i)} x^t_{ijk} \leqslant s^t_{ik}, \quad \forall t \in \mathcal{T}, i \in \mathcal{V}^t, k \in \mathcal{K}. \tag{1b}$$

$$l^t_i \times \sum_{j\in\Delta^+(i)} x^t_{ijk} \geqslant f^t_{ik}, \quad \forall t \in \mathcal{T}, i \in \mathcal{V}^t, k \in \mathcal{K}. \tag{1c}$$

$$s^t_{ik} \leqslant f^t_{ik}, \quad \forall t \in \mathcal{T}, i \in \mathcal{V}^t, k \in \mathcal{K}. \tag{1d}$$

Route structuring

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

$$\sum_{j\in\Delta^+(i)} x^t_{ijk} \leqslant y^t_{ik}, \quad \forall t \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}. \tag{2b}$$

$$\sum_{j\in\Delta^-(i)} x^t_{jik} \leqslant y^{t+1}_{ik}, \quad \forall t \in \mathcal{T} \setminus \{T\}, i \in \mathcal{I}, k \in \mathcal{K}. \tag{2c}$$

$$\sum_{j\in\Delta^+(i)} x^t_{ijk} = y^t_{ik}, \quad \forall t \in \mathcal{T}, i \in \mathcal{N}^t, k \in \mathcal{K}. \tag{2d}$$

$$\sum_{k\in\mathcal{K}} y^t_{jk} \leqslant 1, \quad \forall t \in \mathcal{T}, j \in \mathcal{N}^t. \tag{2e}$$

TS updating

$$R^t_{ijk} \leqslant y^t_{ik}, \quad \forall t \in \mathcal{T}, i,j \in \mathcal{I}, j \neq i, k \in \mathcal{K}. \tag{3a}$$

$$R^t_{ijk} \leqslant y^{t+1}_{jk}, \quad \forall t \in \mathcal{T} \setminus \{T\}, i,j \in \mathcal{I}, j \neq i, k \in \mathcal{K}. \tag{3b}$$

$$R^t_{ijk} \geqslant y^t_{ik} + y^{t+1}_{jk} - 1, \quad \forall t \in \mathcal{T} \setminus \{T\}, i,j \in \mathcal{I}, j \neq i, k \in \mathcal{K}. \tag{3c}$$

$$d^t_{ij} R^t_{ijk} \leqslant D_{\text{limit}}, \quad \forall t \in \mathcal{T}, i,j \in \mathcal{I}, j \neq i, k \in \mathcal{K}. \tag{3d}$$

Car related

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

TS selection

$$z^t_i \geqslant y^t_{ik}, \quad \forall t \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}. \tag{5}$$

Integrity and non-negativity

$$s^t_{ik}, f^t_{ik} \geqslant 0, \quad \forall t \in \mathcal{T}, i \in \mathcal{V}^t, k \in \mathcal{K}. \tag{6a}$$

$$x^t_{ijk}, y^t_{ik}, R^t_{ijk}, z^t_i \in \{0,1\}, \quad \forall t \in \mathcal{T}, i,j \in \mathcal{V}^t, k \in \mathcal{K}. \tag{6b}$$

This model generalizes the MCPRP by Keskin *et al* (2012) from single depot to multi depot and from a static-depot location to dynamic-depot locations. In Constraints (2b)–(2c), y^t_{ik} is forced to 1 when state trooper k starts or his shift at TS i in period t. In the latter case, it is also the starting TS for period $t+1$. The main modification to the formulation involves the newly added Constraints (3a) through (5). The third set of Constraints (3a)–(3c) is confining $R^t_{ijk}=1$, if and only if $y^t_{ik}=y^{t+1}_{jk}=1$, that is, we relocate a state trooper from one TS to another. If relocation occurs, the distance between the starting and the stopping TS should not exceed D_{limit}, which is achieved by Constraint (3d). This is a practical constraint required by the state troopers. In the fourth set, Constraint (4) stipulates that only one car can be parked at one TS. Constraint (5) guarantees that TS is marked as open if selected. Note that if the fixed cost of selecting the TS is 0, that is, $F_i=0$ for all $i \in \mathcal{I}$, then we do not need the variables z^t_i and Constraint (5).

3.1.4. Overall model. The overall model is subject to Constraints (1a)–(6b). We call this model the dynamic multi-depot MCPRP, in short DMD-MCPRP.

Remark 1 If a state trooper must go back to where he starts his shift, R^t_{ijk} and its related constraints are no longer needed. This is a special case of DMD-MCPRP, which can be solved by period and independently.

4. Solution approaches

We observe that DMD-MCPRP is a mixed integer linear program (MILP) and can be solved by CPLEX 12.1.[1] Unfortunately, for the realistic instances of the problem, CPLEX runs out of memory or exceeds the time limit set before it can close the gap between the lower and upper bounds.

[1]CPLEX is a trademark of IBM.

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) using the list of potential TS, and then the exact locations of the TS are determined. This decomposition makes the multi-depot MCPRP problem solvable period by period. Therefore, it provides an opportunity to utilize the solution of Keskin *et al* (2012) 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 without TS location costs and then present a modified heuristic to handle the TS fixed costs.

Box 1: *DMD-MCPRP heuristic* (*Obj**, *Cost**, *Rou**, *Car$_t$**)

1. **Initialization:** $Obj^* = \infty$, $Cost^* = 0$, $Rou^* = \varnothing$, $Car_t{}^* = 0$. Equally allocate $Car^*_t = \min\{\lfloor(1-p)B/(Tv)\rfloor, |\mathcal{K}|\}$ cars $\forall t \in \mathcal{T}$. Select a starting strategy among STR1, STR2, and STR3 for each of the $Car_t{}^*$ cars in period t, $t \in \mathcal{T}$.
2. **MCPRP**[*Obj**, *Cost**, *Rou**].
3. **Add/Drop**[*Obj**, *Cost**, *Rou**, *Car$_t$**].
4. **Select TS**[*Obj**, *Cost**, *Rou**, *Car$_t$**].
5. **if** Budget allows **then**
6. **Insert** [*Obj**, *Cost**, *Rou**];
7. **else**
8. **Erase**[*Obj**, *Cost**, *Rou**, *Car$_t$**].
9. **end if**
10. **Simulated annealing** [*Obj**, *Cost**, *Rou**, *Car$_t$**].
11. Return *Obj**, *Cost**, *Rou**, *Car$_t$**.

4.1. Heuristic for DMD-MCPRP

If there are no fixed costs associated with TS or fixed costs are negligible, the selection of the optimal TS locations depends on the proximity of the TS to the first and final HSs visited in the patrol routes. More specifically, the optimal solution has the following characteristic:

Observation 1 Given the optimal patrol sequence for visiting HSs in MD-MCPRP, the optimal TS locations are the nearest TS to the last HSs in this period and the first HSs in the next period. That is, for a state trooper k and time t, $TS^*_k = min\{i \in \mathcal{I}, d_{i,i^{t-1}} \leqslant D_{limit} : d_{i,j^t_n} + d_{i,j^{t+1}_1}\}$, where i^{t-1} is the TS in period $t-1$, j^t_n is the last HS visited by k in period t, and j^{t+1}_1 is the first HS in period $t+1$.

The main reasons behind Observation 1 are the total cost associated with patrolling and time window considerations of HSs. As an example, let us assume that we have only one patrol car that needs to conduct two shifts. In addition, assume that the optimal patrol sequence information that maximizes the HS coverage is known. At the beginning of Period 1, if we do not select the nearest TS, say i^1, to the first HS, we would be incurring extra mileage costs, and on top of that we may not be able to achieve the desired coverage due to not visiting the first HS on time or within the time window. At the end of Period 1, the car needs to stop at a TS that is reachable from the last HS of Period 1, j_n^1, and first HS of Period 2, j_n^2, with the shortest possible distance travelled. Considering all of the possible TS within D_{limit} miles of the selected TS i^1, it is optimal to select the TS with the shortest travel distance, that is, $TS^* = min\{i \in \mathcal{T}, d_{i,i^1} \leqslant D_{limit} : d_{i,j_n^1} + d_{i,j_1^2}\}$. Otherwise, we would incur additional travel costs and may compromise the coverage.

Observation 1 helps us build our heuristic approach. Note that the first problem is the multi-depot MCPRP that determines the multi-car routing among HSs to maximize the benefits of visiting HSs. This problem ignores the selection of locations for TS 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; and (iii) *STR3*: use a combination of STR1 and STR2, that is, out of the first five earliest HSs, pick the HS with the highest weight. Once a strategy is selected, it is implemented sequentially trooper by trooper to ensure different start locations. The heuristic is run using one of these strategies, and we report the computational results with different strategies in Section 5.

The heuristic has six components: *initialization*, *MCPRP algorithm* (Keskin *et al*, 2012), *add/drop*, *select TS*, insert/erase, and *SA*. The pseudo-code of the algorithm that explains how these components are utilized is given in Box 1. Next, we explain the details of each component.

4.1.1. Initialization. In the Initialization step, we first initialize the objective coverage *Obj**, the total cost *Cost**, 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 troopers with the total number of cars. Initially, *Cost** is set as 0 and is increased as state troopers are utilized and routes are built.

When there are no fixed costs for TS, the operational budget $\mathcal{B}$ is used to pay for the salaries of troopers who are sent on routes and routing cost associated with the gas and mileage. There is an implicit trade-off when it comes to deciding to split the budget against these two components. If all of the budget is spent on salaries, then there will not be any money left for patrolling. On the other hand, if all of the money is spared for patrolling, we will not be able to pay for troopers to be on the roads. We assume that p percent of the budget is initially spent on travel costs related to patrolling and the remaining (1–p) percent is used for paying the salaries of the state troopers.

Initially, to appropriately allocate the budget per car, we assume that in each period we have an equal number of cars. Therefore, over all periods, the total amount of money available for troopers is $(1-p)\mathcal{B}$. When we consider equal split over each period and the cost of a state trooper v, the total number of state troopers employed (cars sent) cannot exceed $\lfloor (1-p)\mathcal{B}/(Tv) \rfloor$, where T is the maximum number of periods as mentioned earlier. Then, the initial number of available cars in period t is calculated as min $\{\lfloor (1-p)\mathcal{B}/(Tv) \rfloor, |\mathcal{K}|\}$, where $|\mathcal{K}|$ is the maximum number of cars available. Afterwards, using one of the aforementioned starting strategies sequentially, 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* (2012) to determine the best patrol routes per car in terms of maximizing HS coverage benefit. This algorithm builds Car_t* routes in a greedy fashion, improved with exchange and relocate operators.

4.1.2. Add/drop component. After the MCPRP algorithm, the initial routes are built for all periods. Using this information, we calculate the total cost *Cost** by taking into account the travel costs incurred by the formed routes. At this point, we need to check whether or not the total cost exceeds the budget. If the total cost exceeds the available budget by more than v, that is, $Cost^* > \mathcal{B} + v$, we eliminate the route with the least coverage time until the total cost is under budget limit. If $\mathcal{B} < \mathcal{B} + v$, we eliminate this infeasibility in the *Insert/Erase Component* that deals with eliminating HSs from patrol routes.

If the total cost is less than the budget, we need to check whether there are enough resources in the budget to create an additional route. We approximate the cost of an additional route as $v + vp/(1-p)$ where the first part is the trooper salary per period and the second part is an approximation of the patrol costs. The approximation of the patrol costs per route is derived as follows: The total budget $\mathcal{B}$ has two components as explained earlier: (i) State trooper salary $= v \times m$, where m is the total number of routes; and (ii) Gas consumption cost $= g \times m$, where g is the approximate gas consumption cost per state trooper. Hence, $\mathcal{B} = m \times (v + g)$. Since we assume that p percent of the budget is for gas consumption and (1–p) percent is for state trooper salaries, we can write the following expression:

$$\frac{m \times g}{m \times v} = \frac{p \times \mathcal{B}}{(1-p) \times \mathcal{B}}.$$

With some simple algebra, we can isolate the approximate gas consumption cost in terms of v as $pv/(1-p)$. Therefore, when a new state trooper is added to the solution, we allocate enough resources (in terms of money) for the salary of the trooper v and for the cost of patrolling $g = pv/(1-p)$.

If *Cost** is less than $\mathcal{B} - (1 + p/(1-p))v$ 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, that is, $Car_t \leftarrow Car_t + 1$. 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 (Keskin *et al*, 2012).

4.1.3. Selecting TS locations. The next step in the overall algorithm involves selecting TS locations. At this point, we have the patrol routes for each state trooper, that is, which HS to visit, in which order, and when. As mentioned earlier, we utilize Observation 1 and simply pick the closest TS to the last HS of the current period and first HS of the next period so that selected TS are within D_{limit} distances of each other. We repeat this process for all $t \in \mathcal{T}$. In essence, this step achieves the goal of picking a common TS that has the smallest travel distance from the final HS of one period to the first HS of the next period in a myopic fashion. After all of the TS locations are selected, the patrol routes are complete. The pseudo-code for selecting TS is given in Box 2.

Box 2: Selecting TS (Cos_t*, Car_t*, *Rou**)

$t \leftarrow 0$.
for Each Car_t in t **do**
  while $t <= T$ **do**
    if $t = 0$ **then**
      Pick the closest TS to the first HSs on routes in period t.
    else
      if $t = T$ **then**
        Pick the closest TS to the last HSs on routes in period T, which is within D_{limit} to the chosen TS in period $T-1$.
      else
        Pick the closest TS to the last HSs on routes in period t and first HSs on routes in periods $t+1$, which is within D_{limit} to the chosen TS in period $t-1$.
      end if
    end if
    $t \leftarrow t + 1$.
    Update *Cost**, *Rou**.
  end while
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).

4.1.4. Insert/erase component. As travelling from the selected TS locations to the HSs increases the total cost due to additional gas consumption, the cost of a new solution after the *Selecting TS* component may exceed $\mathcal{B}$. If the budget is exceeded, *Erase* removes the HS with the least coverage time until $Cost^* < \mathcal{B}$. One possible outcome of this operation is that all of the HSs of a route may have been removed. In that case, that route does not cover any HSs other than TS, and therefore this route is eliminated. That is, a state trooper car is now available. The number of cars used in that period is decreased by one and the total cost is reduced accordingly. Since now additional resources are available by as much as v, the *Insert* component is called to insert any uncovered HSs while considering the travel costs as well as the coverage benefit obtained from the inclusion of new HSs.

On the other hand, if the inclusion of travel costs from and to selected TS locations does not exceed the budget limitation, that is, $Cost^* < \mathcal{B}$, the *Insert* component is run to include uncovered HSs until all of the budget is utilized.

4.1.5. SA component. To optimize the patrol routes and TS locations, we construct an algorithm based on SA. 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), including vehicle routing (Golden and Skiscim, 1986; Osman, 1993; Crainic *et al*, 2010; Yu *et al*, 2010; Lin and Yu, 2012). The basic requirements of SA are a neighbourhood structure on the set of feasible solutions and a number of parameters that govern the acceptance or rejection of new solutions generated during the search. In our SA implementation, we exploit previously defined *Add/Drop* and *Insert/Erase* procedures, while the neighbourhood structure is based on the relocation and exchange of HSs.

Box 3: Procedure $SA(Obj^0, Cost^0, Rou^0, Car_t^0, T_0, \alpha, \beta, M, MaxTime)$

$T = T_0$.
$Cost^{new} = Cost^0$; $Rou^{new} = Rou^0$; $Car_t^{\text{new}} = Car_t^0$; $Obj^{new} = Obj^0$.
$Cost^* = Cost^0$; $Rou^* = Rou^0$; $Car_t^* = Car_t^0$; $Obj^* = Obj^0$.
$Time = 0$.
while $Time \leqslant MaxTime$ **do**
  if $Time > 0$ **then**
    Discard chosen TS, and update Obj^{new}, $Cost^{new}$, and Rou^{new}.
    Add/Drop[Obj^{new}, $Cost^{new}$, Rou^{new}, Car_t^{new}].
    Select TS.

10. **if** Budget allows **then**
11. **Insert**[Obj^{new}, $Cost^{new}$, Rou^{new}];
12. **else**
13. **Erase**[Obj^{new}, $Cost^{new}$, Rou^{new}, Car_t^{new}].
14. **end if**
15. **Select TS**[$Cost^{new}$, Rou^{new}, Car_t^{new}].
16. **end if**
17. Call *Metropolis*($Cost^{new}$, Rou^{new}, Car_t^{new}, Obj^{new}, T, M);
18. **if** $Obj^{new} \geqslant Obj^*$ **then**
19. $Cost^* = Cost^{new}$; $Rou^* = Rou^{new}$; $Car_t^* = Car_t^{new}$; $Obj^* = Obj^{new}$.
20. **end if**
21. $Time = Time + M$;
22. $T = \alpha \cdot T$; $M = \beta \cdot M$.
23. **end while**
24. **return**

SA is a randomized search method that improves 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 are called the *annealing schedule*. The quality of the solution is very sensitive to both of these factors. Therefore, SA requires an initial temperature, T_0; a cooling rate, α; a progressive factor, β; the total allowed time for the annealing process, *MaxTime*; and, finally, the time until the next parameter update, M (Sait and Youssef, 1999, pp 53–55). In our implementation, we experimented extensively to find an effective combination of these parameters. The details of our SA metaheuristic are given in Box 3. In this implementation, an initial solution obtained using Steps 1 through 9 of the *DMD-MCPRP* heuristic, Obj^0, $Cost^0$, Rou^0, Car_t^0, is fed to the SA algorithm. The first time we run through the SA loop (at $Time = 0$), Steps 6 through 15 do not make a difference in the new solution. However, the *Metropolis* procedure, the main component of SA, creates a new solution ($Cost^{new}$, Rou^{new}, Car_t^{new}, Obj^{new}) by simulating the annealing process at a given temperature T. The Metropolis procedure modifies the existing HS routes using exchange and relocate neighbourhoods, as in Keskin *et al* (2012); however, it does not modify the number of routes or the selection of TS. Therefore, when the main loop of SA iterates one more time, it is necessary to check for improvements through *Add/Drop*, *Insert/Erase*, and *Select TS* components. The details of the Metropolis procedure are given in Box 4. Specifically, the existing routes are modified using exchange and relocate neighbourhoods to insert additional HSs or move them to other routes. While searching the neighbourhoods, as opposed to an exhaustive search of the whole neighbourhood, we accept the ‘first best solution.’ That is, we do not seek the best solution in the neighbourhood of the current solution but we accept the first solution that is better than the current solution. If the objective value of the newly generated solution is better than the objective value of the current best solution, we update the best solution (Steps 4–6). Otherwise, we may keep the newly constructed solution and set it as the current solution of the next iteration if the random acceptance rule is satisfied. Earlier in the Metropolis procedure, we have a tendency to accept solutions that are not better than the incumbent as the current solution for the next iteration due to the high value of the temperature parameter. However, as the number of iterations increases, the acceptance rate of non-improving solutions drops.

Box 4: Procedure *Metropolis*($Cost^{current}$, $Rou^{current}$, $Car_t^{current}$, $Obj^{current}$, T, M):

1. **while** $M > 0$ do
2. **Relocate & exchange operator**[Obj^{new}, $Cost^{new}$, Rou^{new}, Car_t^{new},].
3. $\Delta Obj = Obj^{new} - Obj^*$.
4. **if** $\Delta Obj \geqslant 0$ then
5. $Rou^* = Rou^{new}$; $Car_t^* = Car_t^{new}$; $Cos^* = Cost^{new}$; $Obj^* = Obj^{new}$.
6. **else**
7. **if** $Random < exp(-(\Delta Obj)/T)$ **then**
8. $Rou^{current} = Rou^{new}$; $Car_t^{current} = Car_t^{new}$; $Cost^{current} = Cost^{new}$; and $Obj^{current} = Obj^{new}$.
9. **end if**
10. **end if**
11. $M = (M-1)$.
12. **end while**
13. **return** Obj^*, Res^*, $Cost^*$, Car_t^*.

4.2. *Modification of the heuristic for handling fixed costs of TS*

When $F_i > 0$, $i \in \mathcal{I}$, we revise the *DMD-MCPRP* heuristic to handle the fixed cost of TS. Specifically, the inclusion of fixed costs changes two main components of the algorithm. First, instead of locating the TS locations based on proximity to the first and last HSs in the route, we utilize a cost-based approach. For this purpose, we first calculate a cost for each TS based on travel and fixed costs from and to HSs. Next, we select the TS that have the lowest cost. While calculating the costs, we consider the last HS j on each route for each state trooper (ie, every route). We calculate the total travel cost from HS j to a potential TS i (cd_{ij}^t) and increment it with the fixed cost of the TS, F_i. It is critical that the selected TS in consecutive periods are within D_{limit} of each other. Once the costs are calculated, we select the TS locations with the smallest $cd_{ij}^t + F_i$ from the list of potential TS locations that conform to D_{limit}.

Second, since *Add/Drop* more aggressively adjusts the total cost by changing the number of routes Car_t, the algorithm moves onto the modification of TS locations after the total cost exceeds the total available budget. To improve the selected TS locations, we include a *Decrease TS* component that adjusts the total cost less aggressively by dropping one open TS at a time until total cost, *Cost*, drops to $\mathcal{B}$. The rest of the algorithm, including the SA component, stays intact.

5. Computational experiments

In order to test the proposed models and solution approaches, we design small- to medium-sized 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 Critical Analysis Reporting Environment (CARE), a data analysis software package developed by 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 T, the number of HSs per period $|\mathcal{N}|$, the number of state trooper posts $|\mathcal{A}|$, the number of cars $|\mathcal{K}|$, and the number of TS $|\mathcal{I}|$. Even though the number of state trooper posts is not explicitly modelled in the formulation, it affects the problem size and scope due to two main reasons: (i) the state trooper posts are a subset of the potential list of TS; and (ii) the number of state troopers we employ depends on the number of state trooper posts, that is, there is a typical proportionality rule. That is, if there are more state trooper posts, there should also be proportionally more cars $|\mathcal{K}|$, and likewise more $|\mathcal{I}|$. We also assume that $|\mathcal{I}|$ is correlated with T and $|\mathcal{N}|$. For our experiments, we used only 1 day, and therefore the changes in T reflect the changes in the number of shifts in a given day. Once the number of HSs is 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 $2^5 = 32$ instances. The details are provided in Table 1.

On the basis of 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 p: 0.25, 0.5, and 0.75; and
- five budget levels: 20%$\mathcal{B}$, 40%$\mathcal{B}$, 60%$\mathcal{B}$, 80%$\mathcal{B}$, and 100%$\mathcal{B}$, where $\mathcal{B}$ is the estimated maximum budget level. In estimating $\mathcal{B}$, we account for the salary costs and routing costs since we do not have a good reference value. This calculation assumes that we utilize all of the available state troopers in every period, and hence pay the maximum possible salary. To estimate the patrolling (routing) costs, we determine the farthest TS from each HS and assume that the distance from that TS to the HS is travelled twice (in a 'straight-and-back' fashion). Since the triangular inequality among locations holds, this is an upper bound on the distance travelled. Finally, the two components (salary costs and routing costs) are added up to estimate an upper bound on the budget.

Table 1 Design of experiment

Item	*Small*	*Medium*
T	2	4
$\lvert\mathcal{N}\rvert$	16	32
$\lvert\mathcal{A}\rvert$	2	3
$\lvert\mathcal{K}\rvert$	$2\lvert\mathcal{A}\rvert$	$3\lvert\mathcal{A}\rvert$
$\lvert\mathcal{I}\rvert$	$1/8\,T\times\lvert\mathcal{N}\rvert$	$1/4\,T\times\lvert\mathcal{N}\rvert$

For a particular weight scheme, we choose w_j^t for each HS in each period randomly among the three possible values. In total, we run $32\times2\times3\times3\times5=2880$ instances. However, note that CPLEX is run only $32\times2\times5=320$ times, since starting strategies and routing cost allocation percentage levels are heuristic-related parameters. We conduct all of these experiments using C++ on an Intel Core 2 Duo E8400 with 2.94GB of memory. In our experiments, we set the speed limit, φ, as 60 miles/hour. Our proposed metaheuristics return the coverage benefit under the given budget limit. For SA, we set $T_0=1000$, $\alpha=0.9$, $\beta=2$, $Maxtime=8000$ computer seconds, and $M=2$ after extensive experimentation. Meanwhile, since our model is an MILP, CPLEX is able to generate a feasible solution as a benchmark to heuristics. Note that in many instances CPLEX could not solve the problem to optimality within a 3600-second time limit, and hence the reported feasible solution is just a lower bound (LB) to the true optimal value of the objective function. In fact, in many instances, the branch-and-bound tree created by CPLEX exceeds the available memory of the computer even before the time limit. Therefore, it is possible that the heuristics developed are better than the LB of the CPLEX.

5.1. Experimentation for DMD-MCPRP without fixed costs

After obtaining the coverage objective from the heuristic and the LB from CPLEX, we evaluate our solution approach by examining the gap: ($Objective-CPLEX$)/$CPLEX\times100$). 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, that is, the gap is negative. We report both the average and maximum gap, in short 'Avg.' and 'Max.', over all of the experiments in Table 2, as well as the average run times of our heuristic (H. Time) in comparison with the CPLEX time (CT).

In Table 2, the top part of the table is for weights randomly drawn from the low-variance option, and the bottom part is when the weights are randomly drawn from the high-variance option. Comparing different budget levels, there is a general

Table 2 Performance gap between the metaheuristic and CPLEX with average runtimes

	Avg. (%)				*Max.(%)*				*H. Time(s.)*			*CT(s.)*
	Str1	*Str2*	*Str3*	*Best*	*Str1*	*Str2*	*Str3*	*Best*	*Str1*	*Str2*	*Str3*	
					$w_i^t=(1, 1.1, 1.2)$							
$p=0.25$												
100%$\mathcal{B}$	2.3	0.8	1.4	2.6	22.2	20.9	18.9	22.2	0.38	0.38	0.37	2067.7
80%$\mathcal{B}$	1.7	0.2	0.8	2.0	18.0	14.8	16.0	18.0	0.38	0.37	0.37	1740.8
60%$\mathcal{B}$	1.6	0.1	0.8	2.0	20.5	19.2	17.3	20.5	0.60	0.37	0.37	2142.8
40%$\mathcal{B}$	0.4	−1.6	−0.7	0.8	21.5	19.9	18.1	21.5	0.36	0.36	0.35	2065.1
20%$\mathcal{B}$	−1.4	−5.0	−2.8	−0.6	55.9	53.3	55.9	55.9	0.30	0.30	0.66	3149.2
$p=0.5$												
100%$\mathcal{B}$	2.3	0.8	1.4	2.6	22.2	20.9	18.9	22.2	0.40	0.37	0.38	2067.7
80%$\mathcal{B}$	1.7	0.2	0.8	2.0	18.0	14.8	16.0	18.0	0.39	0.37	0.38	1740.8
60%$\mathcal{B}$	1.2	−0.5	0.5	1.7	20.5	19.2	17.3	20.5	0.39	0.37	0.38	2142.8
40%$\mathcal{B}$	−1.6	−2.2	−1.4	−0.2	21.5	19.9	18.1	21.5	0.38	0.36	0.37	2065.1
20%$\mathcal{B}$	−3.7	−4.8	−4.6	−2.2	57.0	57.9	58.5	58.5	0.31	0.29	0.30	3149.2
$p=0.75$												
100%$\mathcal{B}$	1.6	−0.1	0.8	2.0	22.2	20.9	18.9	22.2	0.39	0.37	0.38	2067.7
80%$\mathcal{B}$	−0.5	−1.2	−0.4	0.6	18.0	14.8	16.0	18.0	0.39	0.37	0.38	1740.8
60%$\mathcal{B}$	−1.9	−1.5	−0.8	0.2	17.7	16.0	16.5	17.7	0.40	0.38	0.39	2142.8
40%$\mathcal{B}$	−4.6	−3.7	−3.6	−2.4	19.4	18.8	18.5	19.4	0.39	0.36	0.38	2065.1
20%$\mathcal{B}$	−6.7	−6.6	−5.5	−3.5	54.8	57.9	56.7	57.9	0.29	0.27	0.33	3149.2
					$w_i^t=(1, 1.5, 2)$							
$p=0.25$												
100%$\mathcal{B}$	0.6	−0.4	0.0	1.2	22.8	19.7	19.6	22.8	0.41	0.38	0.40	1775.8
80%$\mathcal{B}$	1.1	0.0	0.4	1.6	23.4	20.2	20.2	23.4	0.41	0.37	0.40	2260.6
60%$\mathcal{B}$	0.6	−0.4	0.0	1.2	13.6	11.1	10.4	13.6	0.41	0.37	0.40	2164.7
40%$\mathcal{B}$	0.9	−0.5	−0.2	1.4	21.3	20.0	19.3	21.3	0.40	0.39	0.38	1800.7
20%$\mathcal{B}$	−0.2	−3.7	−1.7	0.9	56.1	56.1	51.5	56.1	0.31	0.29	0.46	2403.7
$p=0.5$												
100%$\mathcal{B}$	0.6	−0.4	0.0	1.2	22.8	19.7	19.6	22.8	0.42	0.36	0.41	1775.8
80%$\mathcal{B}$	1.1	0.0	0.4	1.6	23.4	20.2	20.2	23.4	0.42	0.37	0.42	2260.6
60%$\mathcal{B}$	0.6	−0.6	−0.3	1.0	13.6	11.1	10.4	13.6	0.41	0.37	0.41	2164.7
40%$\mathcal{B}$	−0.5	−1.0	−0.3	0.6	21.3	20.0	19.3	21.3	0.39	0.36	0.40	1800.7
20%$\mathcal{B}$	−3.1	−2.9	−2.5	−0.4	60.0	53.7	53.5	60.0	0.29	0.27	0.32	2403.7
$p=0.75$												
100%$\mathcal{B}$	−0.1	−1.1	−0.8	0.5	22.8	19.7	19.6	22.8	0.40	0.36	0.39	1775.8
80%$\mathcal{B}$	−0.6	−0.8	−0.4	0.6	23.4	20.2	20.2	23.4	0.40	0.36	0.40	2260.6
60%$\mathcal{B}$	−2.3	−1.4	−1.7	−0.3	13.1	12.1	14.1	14.1	0.41	0.36	0.38	2164.7
40%$\mathcal{B}$	−4.5	−2.0	−2.5	−1.0	15.5	18.4	20.0	20.0	0.37	0.36	0.38	1800.7
20%$\mathcal{B}$	−3.8	−3.2	−3.7	−1.1	52.1	52.5	56.7	56.7	0.29	0.28	0.28	2403.7

trend: as the budgets become tighter and tighter, the average gaps become slightly worse. At 100%$\mathcal{B}$, 80%$\mathcal{B}$, and 60%$\mathcal{B}$, with the 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. The inclusion of the budget limit allows us to conduct the benefit–cost trade-off analysis; the results show how a change in the budget affects the patrol effectiveness. In our experiments, even though this is not shown explicitly in Table 2, we observe that the different sizes of the problem (small *versus* medium) as derived in the experimental design do not have much of an impact on the difficulty level of the problem. The main driver behind these results is the distribution of HS on the map for both our heuristics and CPLEX. A small problem with widely scattered HS is as hard as, or sometimes even harder than, a medium-sized problem with clustered HSs.

Next, we compare different starting strategies from Table 2. Both with lower and higher variance weights, *Str1* has the best average gap for $p=0.25$ with all budget levels and for $p=0.50$ with most budget levels. However, for $p=0.75$, there is no consistent result with respect to which one is the best. For instance, with lower variance weights, *Str3* has the best average gap for most budget levels except 100%$\mathcal{B}$, and with high variance weights *Str2* has the best average gap for budget levels 60%$\mathcal{B}$, 40%$\mathcal{B}$, and 20%$\mathcal{B}$. Because of this lack of consistency, we recommend initializing the heuristics with all of the starting strategies sequentially and proceeding with the best one. Since the heuristic is running fast, this does not create additional problems.

Table 3 Performance gap between the 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.(%)*			
	Str1	*Str2*	*Str3*	*Best*	*Str1*	*Str2*	*Str3*	*Best*	*Str1*	*Str2*	*Str3*	*Best*	*Str1*	*Str2*	*Str3*	*Best*
$p = 0.25$																
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
$p = 0.5$																
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
$p = 0.75$																
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

Our third finding from the analysis indicates that the initial route cost allocation factor, p, does not affect the coverage benefit. In the first two rows with $p = 0.25$ and those with $p = 0.5$, the results are exactly the same. This is due to the heuristic's ability to effectively modify the resource usage among gas consumption and state trooper cars. Regardless of how much budget is allocated to gas consumption in the beginning of the algorithm, the inherent *Add/Drop* component alters the number of cars very effectively. We, in fact, conduct an indicator variable analysis using Minitab 16.1.1 by taking the 'best' averages as responses. According to the analysis, there is no statistical difference between two different p values. The performance of the heuristic is also very similar with different w_i^t in terms of coverage benefits and run times. The heuristic performs slightly better on low-variance weights when the budget is not very restrictive (ie, 100%$\mathcal{B}$, 80%$\mathcal{B}$, and 60%$\mathcal{B}$). However, it performs better on high-variance weights when the budget is tighter (ie, 40%$\mathcal{B}$ and 20%$\mathcal{B}$). The sound performance of our approach with regard to different criteria 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 five factors have positive impacts on the objective. The positive relation between the number of TS and the coverage objective forms the root cause of the necessity to incorporate the choice of TS in the patrol routes of state troopers.

As for runtime, as can be seen in Table 2, our metaheuristic takes less than a second on average, while the run time of CPLEX is more than 2000 seconds to obtain a feasible solution. Therefore, our solution approach received favourable feedback from the state troopers.

5.2. Experimentation for DMD-MCPRP with fixed costs

Next, we investigate the performance of the revised metaheuristic to solve the model with $F_i > 0$. Keeping all other parameters the same, we test our algorithm with an 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 that case as well. Since weight scheme $(1, 1.1, 1.2)$ and weight scheme $(1, 1.5, 2)$ have very similar results, we only report the results of one weight scheme $(1, 1.1, 1.2)$ to avoid redundancy. In Table 3, we report average gaps for 32 experiment instances compared with CPLEX.

When we compare the results for different budget levels, starting strategies, and p values, 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 tighter the budget and the higher the fixed cost of TS, the more

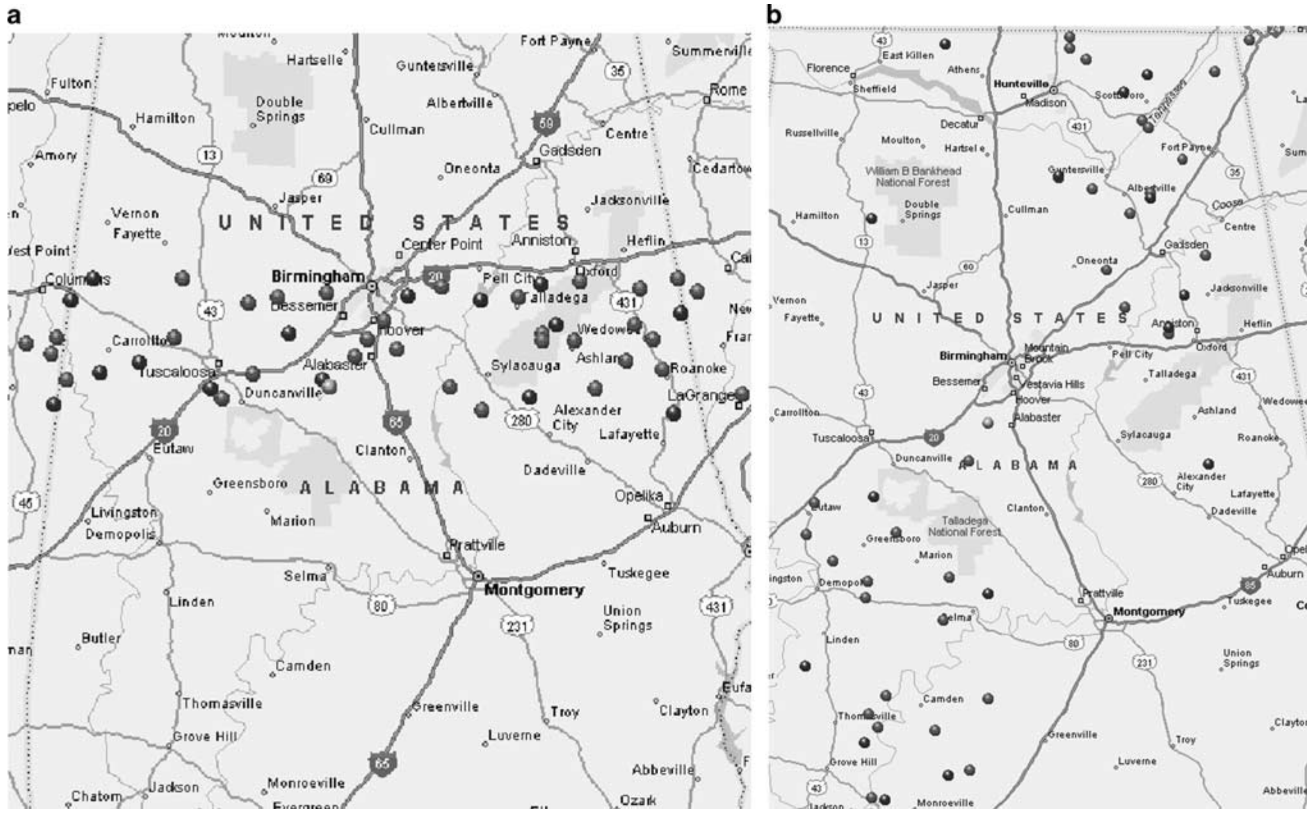

Figure 2 HSs of uniform distribution and clustered distribution.

obvious this phenomenon is. It can be projected that if the budget is very tight and the fixed cost of TS is high enough, all state troopers will share only one TS in each period, which becomes a single-depot problem.

5.3. Performance measures

In addition to comparing our heuristic with CPLEX, we also perform a benchmark analysis against the algorithm in Keskin *et al* (2012), since DMD-MCPRP is an extension of MCPRP. In order to run the algorithm by Keskin *et al* (2012), we assume that a central state trooper post, a facility from the potential TS list, acts as the single depot, and that all of the state troopers are dispatched from this location every period.

Other than the total coverage time objective, Keskin *et al* (2012) also introduce 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 following definitions:

HS%: This performance measure calculates, among all of the HSs, the percentage covered as a result: $HS\% = \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}}\sum_{k\in\mathcal{K}} y^t_{ik}/(T\times|\mathcal{N}|)$, where the numerator represents the total number of visited HSs.

TW%: This performance measure calculates the percentage of total available time serviced: $TW\% = \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}}\sum_{k\in\mathcal{K}}(f^t_{ik}-s^t_{ik})/\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}}(l^t_i-e^t_i)$. 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 p and starting strategies of DMD-MCPRP are compared with objectives of MCPRP returned by local search. On the basis of the previous experiment design, we set the number of TS $|\mathcal{I}|$ equal to the product of T and $|\mathcal{N}|$, and thus $|\mathcal{I}|$ is not a factor in the new design. In total, there are 16 instances for comparisons. Since our solution approach performs in a similar fashion with respect to different weight schemes, we only focus on the low weight (1, 1.1, 1.2) in reporting the results.

In order to better demonstrate the differences between DMD-MCPRP and MCPRPR, we focus on two different geographical spreads of HSs, which are plotted in Figure 2. The red dots represent HSs, the blue dots represent the potential TS including state trooper posts, and the green dot represents the single depot. In Part (a), HSs are spread within the centre of the state (centralized HSs), and in Part (b), HSs are distributed throughout the state (scattered HSs). The single depot is located in the centre of the state, and it is closer to HSs in Part (a). We test and compare the outcomes of DMD-MCPRP with those of MCPRP using both centralized and scattered distributions.

Table 4 Comparison of performances between MCPRP and DMD-MCPRP with centralized distribution

			MCPRP			*DMD-MCPRP-20*				*DMD-MCPRP-240*			
Ins	*TotTm*	*TotHS*	*Obj*	*TW%*	*HS%*	*Obj*	*TW%*	*HS%*	*Imp%*	*Obj*	*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
Average				79	91		80	91	3.3		81	91	3.9

We first present the results for centralized HSs distribution in Table 4. This table includes the following columns: the data instance from the experimental design ('Ins'), total possible coverage time ('TotTm'), total number of HSs available ('TotHS'), objective value in terms of total coverage time ('Obj'), and improvement over the objective in percentage ('Imp'). We also vary D_{limit} to show its impact on the performance of our solution approach. We set D_{limit} to 20 and 240 to show how restrictive D_{limit} can be. To compare objectives of MCPRP and DMD-MCPRP when $D_{limit}=20$, we calculate the improvements in column 'Imp' as ('Obj' of 'DMD - MCPRP - 20' – 'Obj' of MCPRP)/ 'Obj' of MCPRP × 100 and when $D_{limit}=240$, we calculate the improvements in column 'Imp' as ('Obj'of 'DMD - MCPRP - 240' – 'Obj' of MCPRP)/ 'Obj' of MCPRP × 100. The results confirm our intuition that DMD-MCPRP outperforms MCPRP.

For $D_{limit}=20$, based on the results from 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-20, on average, has 3.3% more time coverage benefits than MCPRP. Even though this percentage may seem low, in practice this translates to approximately 2556 min of effective coverage. The TW% performances of both MCPRPR and DMD-MCPRP-20 are consistent with total coverage time objectives, as they have the same denominators. On average, DMD-MCPRP-20 returns 80% and MCPRP returns 79% TW% coverage. Thus, with DMD-MCPRP-20, state troopers stay at HSs longer. With respect to the HS%, there is no relationship between DMD-MCPRP-20 and MCPRP, with both of them resulting in 91% HS% coverage. Thus, with MCPRP, state troopers move more often from one HS to another due to the elapse of the time window. On average, both performance measures are higher than 80%, which is quite satisfactory from the state troopers perspective.

For $D_{limit}=240$, the results are shown in the column 'DMD-MCPRP-240' in Table 4. 'DMD-MCPRP-240' further improves the results of MCPRP in terms of time coverage by an average of 3.9%. It translates into an additional coverage of 2960 min, almost 50 h, over MCPRP. These results show that if we select a larger radius D_{limit}, the objective in terms of coverage time improves. Improvement at such scale helps state troopers increase their patrol effectiveness.

We next present the results for scattered HSs, as in Figure 2(b), in Table 5, which has the same structure as the previous table. 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 $D_{limit}=20$ (a total of 5357 min of additional coverage) and MCPRP by 11.7% when $D_{limit}=240$ (a total of 6668 min of additional coverage). TW% and HS% are improved from 60% and 75% to 64% and 78%, respectively, when $D_{limit}=20$, and further improved to 67% and 78% when $D_{limit}=240$. These results show the importance of geographical spread of HSs in selecting DMD-MCPRP *versus* MCPRP. With both centrally distributed and scattered HSs, DMD-MCPRP outperforms MCPRP. The difference in performance in terms of coverage time, TW%, and HS% is more pronounced for scattered HSs.

Table 5 Comparison of performances between MCPRP and DMD-MCPRP with clustered distribution

			MCPRP			*DMD-MCPRP-20*				*DMD-MCPRP-240*			
Ins	*TotTm*	*TotHS*	*Obj*	*TW%*	*HS%*	*Obj*	*TW%*	*HS%*	*Imp%*	*Obj*	*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
Average				60	75		64	78	9.6		67	78	11.7

6. Conclusions

In conclusion, in order to improve the efficiency of state trooper patrols, we allow for dynamically changing patrol routes and multiple starting and stopping locations for patrol routes. For this purpose, we develop a new dynamic, multi-depot, location-routing model, extending the existing MCPRP in the literature. With 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 with the feasible solutions returned by CPLEX. We also compare the time and HS coverage performances of this model with the single-depot MCPRP and show that significant improvements are possible in terms of coverage time and HS visits.

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

Acknowledgements—The authors thank the University Transportation Center for Alabama for providing support for this research under the grant UTCA-11113. 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. Finally, the authors are grateful for the constructive comments of the two anonymous referees.

References

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

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

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

Balakrishnan A, Ward JE and Wong RT (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 JF, Gendreau M and Potvin JY (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 *et al* (2002). The economic impact of motor vehicle crashes, 2000. Crash Statistics DOT HS 809 446, NHTSA.

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

Caballero R, González M, Guerrero FM, Molina J and Paralera C (2007). Solving a multiobjective 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 and Haimes YY (1983). *Multiobjective Decision Making: Theory and Methodology*. Vol. 8 North-Holland.

Crainic TG, Gendreau M and Rousseau L-M (2010). Special issue on recent advances in metaheuristics. *Journal of Heuristics* **16**(3): 235–237.

Golden B and Skiscim C (1986). Using simulated annealing to solve routing and location problems. *Naval Research Logistics Quarterly* **33**(2): 261–279.

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

Keskin BB, Li SR, Steil D and Spiller S (2012). Analysis of an integrated maximum covering and patrol routing problem. *Transportation Research Part E: Logistics and Transportation Review* **48**(1): 215–232.

Kirkpatrick S, Gelatt CD and Vecchi MP (1983). Optimization by simulated annealing. *Science* **220**(4598): 671–680.

Labbé M, Rodrguez-Martin I and Salazar-González JJ (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. In: Golden BL and Assad AA (eds). *Vehicle Routing: Methods and Studies*. North-Holland, Amsterdam: pp 163–198.

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

Laumanns M, Thiele L and Zitzler E (2005). An adaptive scheme to generate the pareto front based on the epsilon-constraint method. In: Branke J, Deb K, Miettinen K and Steuer RE (eds). *Practical Approaches to Multi-Objective Optimization*. Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

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

Lin S-W and Yu VF (2012). A simulated annealing heuristic for the team orienteering problem with time windows. *European Journal of Operational Research* **217**(1): 94–107.

Lou Y, Yin Y and Lawphongpanich S (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 ε-constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation* **213**(2): 455–465.

Melechovský J, Prins C and Calvo RW (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*. International Series in Operations Research and Management Science, Vol. 12. Springer: New York.

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

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

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

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

Osman IH (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* **41**(4): 421–451.

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

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

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

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

Steil D and Parrish A (2009). Hit: A gis-based hotspot identification taxonomy. *International Journal of Computers and Their Applications* **16**(2): 81–90.

Tricoire F, Romauch M, Doerner KF and Hartl RF (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, Souffriau W, Vanden Berghe G and Van Oudheusden D (2009). Iterated local search for the team orienteering problem with time windows. *Computers and Operations Research* **36**(12): 3281–3290.

Willemse EJ and Joubert JW (2012). Applying min–max k postmen problems to the routing of security guards. *Journal of the Operational Research Society* **63**(2): 245–260.

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

Yu VF, Lin S-W, Lee W and Ting C-J (2010). A simulated annealing heuristic for the capacitated location routing problem. *Computers and Industrial Engineering* **58**(2): 288–299.

Received 9 February 2012;
accepted 20 August 2013 after two revisions