This is the author's version of an article that has been published in IEEE Transactions on Automation Science and Engineering.
DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

Vehicle Routing for Resource Management in Time-Phased Deployment of Sensor-Networks

Kasper Woiceshyn, Zendai Kashino, *Student Member*, *IEEE*, Goldie Nejat, *Member*, *IEEE*, Beno Benhabib

***Abstract*—Time-phased sensor-network deployment refers to the delivery of a set of sensors to their predetermined locations at exact times by a fleet of vehicles. Applications for such network deployments include wilderness search and rescue and wildfire monitoring, where desirable resource management would imply allowing the vehicles to perform other tasks between deliveries. The goal of this paper is, thus, to formulate and solve a vehicle-routing problem for such *just-in-time* time-phased sensor-network deployments.**

The proposed optimization method for the modified vehicle-routing problem outlined herein has two primary novelties: (*i*) the consideration of *spare-time* as the objective function, and (*ii*) the use of a targeted local-search method. The spare-time objective function was formulated to address the uniqueness of the modified routing problem at hand. The targeted local-search algorithm, on the other hand, was developed to tangibly improve the efficiency of the search for the optimal values of the chosen objective function.

The proposed vehicle-route planning method was validated via a range of simulated wilderness search and rescue scenarios, some of which are included herein. The robustness of the method to variations in problem parameters was also investigated.

***Note to Practitioners*—The resource-management problem addressed in this paper is applicable to scenarios wherein a fleet of vehicles visits a set of locations at pre-determined times to provide services while carrying out other tasks in-between. Such time-phased applications include deployment of sensor networks for wilderness search and rescue or wildfire monitoring, patient transportation services that can handle emergencies, and courier services that can cope with urgent express requests**

The primary inputs to the proposed vehicle-routing algorithm are (*i*) the physical characteristics of the vehicles (i.e., speed, capacity, operation time limit, etc.), and (*ii*) a service plan (i.e., service locations and corresponding exact service times). The algorithm yields best possible routes (a string of assigned service locations) for all the vehicles, by maximizing *spare time* between deliveries (within a reasonable computation time). The method also allows for changes in service plan in real-time.

***Index Terms*—Resource management, vehicle routing, wireless sensor networks.**

This research was funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Canada Research Chairs program.

All authors are with the Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON M5S 3G8, Canada. (zendkash@mie.utoronto.ca)

I. Introduction

Wireless sensor networks (WSNs) have been commonly proposed for environmental monitoring, structural health monitoring, search and rescue, and surveillance [1]-[17]. Research in this area, however, has primarily focused on network-configuration planning [18]-[23]. For example, [22], [23] focus on planning deployments to maximize sensor-network lifetime. The topic of network deployment, in terms of how to deliver sensors to their planned locations, is often overlooked. In particular, the delivery of sensors for time-phased network deployment has not been addressed explicitly in the literature.

Time-phased sensor network deployment examples include wilderness search and rescue (WiSAR) and wildfire monitoring. In such applications, the subject-of-interest is dynamic and optimal sensor deployment locations (i.e., network configuration) may vary over time.

In WiSAR, for example, the goal is to locate an un-trackable, moving target (missing person) in an unbounded, expanding area. Prior work for this application has proposed the use of mobile robots with on-board sensors to conduct the target search [24]-[26]. The use of WSNs in support of such search efforts has also been proposed, where sensor deployment locations and times are optimized to maximize likelihood of target detection [7].

In wildfire monitoring, a fleet of autonomous vehicles, such as unmanned aerial vehicles (UAVs), could be dispatched to track the progress of the firefront [27]-[29]. In addition to aerial monitoring, the supplemental use of static sensors has also been considered [30]. The UAVs could deploy the static sensors at optimized locations on an optimized schedule for closely monitoring the moving firefront while also conducting aerial surveys between deployments.

In both above examples, and other similar problems (i.e., time-phased deployment of a network), allowing the vehicles to perform other tasks between deliveries could be desirable. Thus, herein, optimal resource management refers to planning routes that maximize vehicle *spare times* between deliveries.

A. Traditional Vehicle Routing

While vehicle routing has been well studied in the literature, in a typical problem, vehicles are exclusively used for delivery with no idle time. Namely, vehicles move between deliveries as fast as possible, even when time-windows are considered. Such, vehicle-routing problems (VRPs), typically, belong to

DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

the class of NP-hard combinatorial optimization problems [31]. As a generalization of the traveling salesperson problem (TSP), a VRP involves optimally assigning and sequencing a set of customer visits among a fleet of available vehicles [32]-[43].

Most VRPs feature a central depot, from which all routes originate and terminate at. A variant of this is the open VRP, in which routes may end at any location [44], [45]. Furthermore, a multi-depot VRP was addressed in [46], where there are multiple locations from which routes may originate.

Other VRPs have considered variations to the vehicle fleet, temporal constraints, and dynamic nature of the problem. Both homogeneous and heterogeneous vehicle fleet variants [47], [48] have been studied, particularly with respect to capacities, maximum speeds, and operating costs. Customer visits are often time-constrained, either by time windows [49]-[51], or time deadlines [52]. A dynamic version of the VRP has also been studied, in which all yet unfulfilled requests are re-arranged online to optimally accommodate new sporadic requests when they arise [53]. Metrics of optimality in the abovementioned problems have included distance travelled, number of vehicles employed, and arrival times at delivery locations [54]-[61].

The current state-of-the-art methods for solving VRPs include metaheuristic methods that are markedly efficient compared to exact methods on large, highly-constrained problem instances. These have been categorized into single-solution methods and population-based methods. Single-solution methods are those that operate on a single incumbent solution, often through modifications known as moves. Examples of commonly utilized methods in this category include Tabu search [62]-[65], variable neighborhood search [66]-[68], and simulated annealing [69], [70]. Population-based methods are those that operate on a population of incumbent solutions, examples of which include ant colony optimization [71], [72], genetic algorithms [73]-[77], and particle swarm optimization [78].

B. Vehicle Routing for Resource Management

In this paper, a unique resource-management problem, for time-phased network deployments, is addressed. Namely, a fixed-size fleet of mobile vehicles is coordinated to make a series of *just-in-time* deliveries, while maximizing the vehicles' ability to perform other tasks between deliveries. The vehicles are assumed to have heterogeneous capacities, speeds, and starting locations.

The above problem can be formulated as a variant of the traditional VRP. Though, it must consider maximizing a *spare-time* type objective function for optimal route planning. This contrasts with classical VRPs, where the objective is to minimize the cost of deliveries. The new spare-time metric must also have both spatial and temporal components. Past metrics have been, typically, formulated only spatially (e.g., considering total travel distance or fuel consumption), with temporal constraints (e.g., deadlines or windows for delivery) [21]-[62].

A novel solution method to the new variant VRP outlined above is proposed in this paper. Specifically, a targeted local-search method is proposed to effectively maximize spare-time available to vehicles. The targeted search only considers moves that involve replacing the arc with the minimum spare-time value. When compared to traditional local-search techniques, which search the entire local neighborhood of the current solution, a targeted local-search reduces computational complexity by an order of magnitude (i.e., from $O(n^2)$ to $O(n)$).

The new variant VRP was inspired by emerging research on autonomous robotic deployment of sensor networks. However, the proposed solution method can be adapted to other similar applications. Examples include a patient transportation service that also handles emergencies [79], or a courier service that also handles urgent express requests [80].

II. Problem Definition

The variant VRP addressed in this paper considers optimal resource management in the time-phased deployment of sensor-networks. In this problem, a fleet of vehicles is required to deliver a set of sensors to specified deployment locations *just-in-time*. The focus is to maximize the fleet's effectiveness in performing other tasks, while carrying out its primary task of sensor delivery. This is achieved by maximizing the *spare-time* available to vehicles between sensor deliveries.

Maximization of spare time, however, should not unfairly favor any part of the deployment. Thus, in order to achieve an unbiased maximization of the objective function, herein, we propose maximizing the minimum spare time between any two deliveries while considering all vehicles.

A. Time-Phased Sensor Delivery

Let us consider the deployment of n sensors. Each sensor is associated with a deployment location, $\boldsymbol{x}_i$, and time, t_i, $i \in \{1, \cdots, n\}$. The set $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$, thus, denotes the set of all deployment locations and $\{t_1, \cdots, t_n\}$ denotes the ordered set of all deployment times (i.e., $t_1 < \cdots < t_n$). These act as constraints to the route-optimization problem as they cannot be altered. One can note that, even if a vehicle could arrive at a deployment location early, it may not leave before the associated specific sensor-delivery time.

Optimal sensor-deployment locations and corresponding deployment times are determined by an external network-planning methodology. For example, the algorithm outlined in [7] plans sensor locations that maximize the likelihood of target detection in a WiSAR scenario. Corresponding deployment times are determined to spread the sensor-network's search effort uniformly over time.

During vehicle-route planning, the set of all sensor-deployment locations and times are divided into non-intersecting subsets – one for each robot. For example, if a robot is assigned to deploy Sensors 1, 6, and 9, the corresponding subset of deployment locations would be $\{\boldsymbol{x}_1, \boldsymbol{x}_6, \boldsymbol{x}_9\}$ with deployment times $\{t_1, t_6, t_9\}$. One can note that $t_1 < t_6 < t_9$ since, in general, $t_i < t_j$ for $i < j$, $i, j \in \{1, \cdots, n\}$.

DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

Herein, it is assumed that a fleet of K vehicles is available – each with a distinct capacity of sensors that it can deploy, q_k, and a distinct speed, v_k, $k \in \{1, \cdots, K\}$. It is also assumed that the vehicle would maintain a constant speed throughout the execution of the planned route. The vehicles' routes may originate from different locations, $\boldsymbol{x}_{n+k}$, at different times, t_{n+k}.

For the variant VRP, each vehicle may be assumed to have an operating time limit, based on fuel or energy consumption. This could impact route planning by, for example, requiring robots to refuel during sensor deployment. Herein, however, it is assumed that the time of the last delivery, t_n, is chosen to be less than the smallest maximum operating time of any vehicle. This eliminates the need to consider restrictions imposed by a limited robot battery life/fuel capacity.

B. Problem Formulation

The deployment of n sensors by K vehicles can be represented spatially on a Euclidean graph, $G = (N, A)$, where $N = \{1, \cdots, n + K\}$ is the set of sensor deployment nodes ($\{1, \cdots, n\}$) and robot origin nodes ($\{n + 1, \cdots, n + K\}$) and A is the set of arcs connecting them. Each Arc traversed by robot k, $(i, j, k) \in A$, connecting Locations i and j, is associated with a spare time value, τ_{ij}^k. This is the spare time that Vehicle k has before reaching Location j, at Time t_j, having made a delivery at Location i immediately before, at Time t_i. The spare time, τ_{ij}^k, is calculated assuming the vehicle travels the path, from Location i to Location j, with the shortest possible travel time:

$$\tau_{ij}^k = t_j - \left(t_i + \delta_{ij}^k\right), \tag{1}$$

where δ_{ij}^k is the minimum travel time for Vehicle k between Locations i and j. In our work, this travel time is deterministic and a function of distance, terrain, and vehicle speed, v_k.

In a WiSAR scenario, for example, the travel time would be estimated by considering the robot specifications, distance between locations and terrain information. The minimum travel time could, then, be approximated by, using a shortest path algorithm such as Dijkstra's on a discretized terrain map.

Above, a negative τ_{ij}^k value would imply that the vehicle will be late for the delivery of the sensor at Location j, if Arc (i, j, k) were to be taken.

A solution to the routing problem, S, could consist of a selection of n arcs in A. Let a set of binary arc selection variables, x_{ij}^k, be equal to 1, if Arc (i, j, k) is selected in S, or 0, otherwise. Namely, a selection of Arc (i, j, k) implies that Vehicle k travels directly from Location i to Location j in the solution. An optimal solution maximizes the spare-time available, while ensuring that the opportunity for focusing on other tasks is also distributed uniformly. The objective function, f, can, then, be expressed as the minimum τ_{ij}^k selected in S:

$$f = \min_{i,j,k} \tau_{ij}^k, \qquad x_{ij}^k \neq 0. \tag{2}$$

The overall route-optimization problem at hand is, thus, defined as:

$$\text{Maximize } f, \tag{3}$$

subject to the following constraints:

$$\sum_{k=1}^{K} y_i^k = 1, \qquad \forall\, i \in N, \tag{4}$$

$$\sum_{i=1}^{n+K}\sum_{j=1}^{n}\sum_{k=1}^{K} x_{ij}^k = n, \tag{5}$$

$$\sum_{i=1}^{n+K} x_{ij}^k = y_j^k, \qquad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, K\}, \tag{6}$$

$$\sum_{j=1}^{n} x_{ij}^k \le y_i^k, \qquad \forall i \in N, \forall k \in \{1, \dots, K\}, \text{ and} \tag{7}$$

$$\sum_{i=1}^{n} y_i^k \le q_k, \qquad \forall k \in \{1, \dots, K\}. \tag{8}$$

Above, y_i^k, is equal to 1, if Vehicle k visits Location i in S, and 0, otherwise. Eq. (4) represents the constraint that each sensor-deployment location is to be visited by exactly one vehicle, where. Eq. (5) represents the constraint that exactly n arcs are selected in S, defined as $n - K$ delivery-to-delivery arcs, and K arcs from vehicle start locations to their first deliveries. Eq. (6) represents the constraint that each delivery location must be arrived only once. Eq. (7) represents the constraint that each location, whether it is a vehicle start location or a delivery location, must be departed from at most once. The terminal location of each route is not departed from. It is assumed that each vehicle will remain at or near the location of its final delivery. Further instructions could be provided to the vehicle after completing deployment. Eq. (8) represents the constraint that no vehicle can be assigned more deliveries than for which it has capacity.

C. Late Deliveries

A negative objective function value, f, indicates that at least one sensor is deployed late. In this case, an optimal solution per Eq. (2) is one for which the maximum lateness of any delivery is minimized.

If late delivery is not acceptable, one approach that can be taken is to modify the fleet to enable the on-time delivery of all sensors. Alternatively, one could find the largest subset of the first n' sensors that can be deployed on-time (i.e., such that the objective function in Eq. (2) is positive):

$$f > 0, i, j \in \{1, \cdots, n'\}. \tag{9}$$

Also, the formulation of τ_{ij}^k, as per Eq. (1), assumes that Vehicle k leaves Location i on-time. If this is not the case and spare time for the vehicle arriving at Location i is negative, a push-forward must be calculated, if it is desired to determine the actual delivery times resulting from lateness.

D. NP-hardness

The new variant VRP can be shown to be NP-hard by establishing equivalence to a problem known to be NP-hard.

DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

Herein, we choose to show equivalence to the bottleneck travelling salesman problem (BTSP) [61]. The BTSP is solved by minimizing the maximum edge cost in a Hamiltonian tour.

Proof: Let us define the graph on which we would like to solve the BTSP by G. Additionally, an edge cost function is defined such that the cost of going from node u to node v is $c(u, v)$.

Our max-min variant VRP, with a fleet of one vehicle on the graph G and with a cost function of $-c(u, v)$, is equivalent to the original BTSP described above. The max-min VRP instance specifies the problem of finding a Hamiltonian tour of the graph G that maximizes the minimum edge cost $-c(u, v)$ used in the tour. This is equivalent to the original BTSP, since the objective of maximizing the minimum of $-c(u, v)$ in the tour is equivalent to minimizing the maximum of $c(u, v)$ in the tour. The BTSP is, therefore, a special case of our max-min VRP, thus, showing that the max-min VRP is at least as complex as the BTSP. Since the BTSP is known to be NP-hard, this indicates that the max-min VRP being addressed in this paper is also NP-hard.

III. Proposed Resource-Management Method

A novel solution method is proposed herein for the unique variant VRP formulated above, Eq. (2) to (8). Route planning starts with a set of inputs, including a list of sensor deployment locations, corresponding deployment times, and parameters of the available fleet. Fleet parameters may include the number of vehicles, individual capacities, speeds, and maximum operating times, etc.

An illustration of the proposed method is given in Fig. 1.

A. Route-Planning Algorithm

There are a range of approaches and techniques for addressing the route-planning problem at hand, including evolutionary ones that recombine high-quality route plans (parents) to create even higher quality plans (offspring). During our research, we investigated and compared several different approaches to route optimization. These included Simulated Annealing, Ant Colony Optimization, Genetic Algorithm, and Variable Neighborhood Search (VNS) and Tabu Search (TS). The comparisons revealed that the local-search based VNS and TS yielded the best results.

We propose the use of VNS and TS metaheuristic methods to determine vehicle routes. Our novel *targeted local-search* algorithm will be used in these methods for optimal resource management. The proposed algorithm repeatedly finds and removes the arc in the solution representing the lowest spare time, replacing it with arcs of higher value. Typically, local search operates via moves, in which a set of arcs is removed and replaced by a different set of arcs, monotonically improving the objective function until a local optimum is reached. The metaheuristic methods guide the local search from prematurely converging to local optima.

It should be noted, however, that neither TS nor VNS are polynomial-time approximation algorithms and, thus, have no performance or convergence guarantees (e.g., approximation ratios and worst-case computation time). Both methods can run indefinitely, searching the solution space. It can be claimed, therefore, that their theoretical approximation ratio approaches 1, as their search time tends to infinity. Namely, given an unlimited amount of computation time, the probability that the search methods will find the global optimum solution approaches 1. However, this is not a particularly useful result as there is, typically, always a time-critical element to problems like WiSAR. Thus, it would be more instructive to investigate what the approximation ratio is for a given finite-time limit. Such an investigation is carried out in Section IV using realistic simulated experiments.

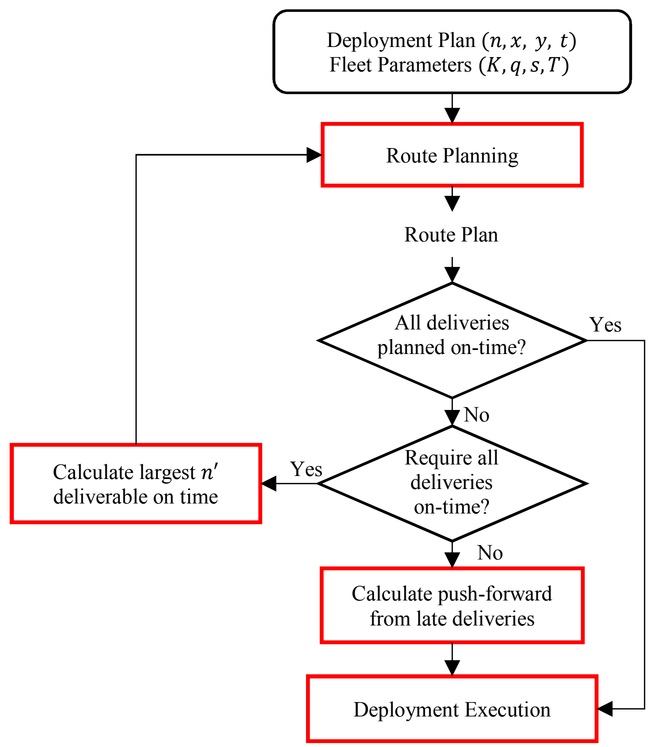


Fig. 1. Overview of the proposed route-planning method.

1) Local-Search Moves

In order to address the objective of *maximizing minimum spare time*, we propose a local search (LS) with three types of moves: swap, relocate, and 2opt*, each of which has a neighborhood of cardinality approximately n^2. Each move is illustrated in Fig. 2. Delivery locations are represented by circles and vehicle starting locations by squares, respectively. Red arcs represent those removed by the move, A_{Rmove}, and green arcs represent those added by the move, A_{Amove}.

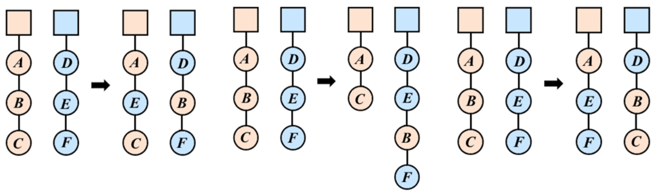
An iteration of a typical LS would involve, examination of every possible move that could be made, computation of the objective function value of each potential move, and the execution of the move with the highest value.

2) Targeted-Local Search

As per Eq. (2), the value of the objective function of a solution (set of routes) is defined by its lowest-value arc.

DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

Therefore, in our novel *target local-search*, we only search for improving moves that remove the lowest-value arc. In this manner, each neighborhood can be evaluated in $O(n)$ rather than $O(n^2)$.



(a) (b) (c)

Fig. 2. Illustration of (*a*) swap of *B* and *E*, (*b*) relocation of *B* after *E*, and (*c*) 2opt* move between *A-B* and *D-E*.

As an example, let us consider the route given in Fig. 3, which contains the lowest-value arc in the solution, (B, C, k). A single iteration of targeted swap LS considers every pair of locations containing B or C, and every other location Y in the remaining routes. Namely, potential moves are those in which Y is exchanged with B, or Y is exchanged with C. An iteration of targeted relocate LS, similarly, considers moves in which B is relocated after Y, C is relocated after Y, and Y is relocated after B. An iteration of targeted 2opt* LS evaluates moves in which the tails of B and Y are exchanged. Pseudocode describing each type of LS is given in Appendix A.

Fig. 3. Route with lowest-value arc (B, C, k).

Evaluating the quality of a move can also be simplified when using targeted LS. Namely, since potential moves are restricted to those that remove the lowest-value arc, the value of a move can be computed as the minimum value of the added arcs:

$$f_{MMmove} = \min_{(i,j,k)\in A_{Amove}} \tau_{ij}^{k}. \tag{10}$$

This reduces the computational complexity of move evaluation from $O(n)$ to $O(1)$ since only a fixed number of arcs are added for every move made.

3) *Variable-Neighborhood Search*

In the method proposed in this paper, variable-neighborhood search (VNS) is one of two metaheuristic methods within which the targeted local search was embedded. In the implementation described here, VNS involves a variable neighborhood descent (VND) subroutine that operates as follows:

1. Perform LS with the swap move until no further improvement is possible.
2. Perform LS with the relocate move until no further improvement is possible. If any improvement is made by a relocate move, return to Step 1. Otherwise, continue to Step 3.
3. Perform LS with the 2opt* move until no further improvement is possible. If any improvement is made by a 2opt* move, return to Step 1. Otherwise, exit.

First, the swap neighborhood is explored, as it maintains the same number of locations assigned to each route and, thus, does not violate the constraint in Eq. (8). Then, the relocate neighborhood is explored before the 2opt* neighborhood, as the latter reassigns a larger number of locations between routes. Finally, each iteration of VND is alternated with a shaking move, consisting of one of:

- a randomly chosen move from one of the three LS neighborhoods, or
- a random 3opt* move, in which the tails of three routes are exchanged, or
- a 4-relocate, in which four locations are removed and re-inserted at random.

Iterations of VND and shaking are repeated until a stopping condition, such as a time limit, iteration limit, or convergence criterion, is met.

4) *Tabu Search*

Tabu search (TS) is another metaheuristic method used to evaluate the performance of the targeted local search. It is implemented as a fully deterministic method, relying on a memory structure to avoid premature convergence, as opposed to randomization. In our proposed TS, local search is carried out by exploring the three LS neighborhoods (swap, relocate, and 2opt*) before taking the highest-value non-Tabu move.

After every move made by the TS, arcs involved in the move are marked Tabu, or prohibited, for a limited number of iterations, known as the Tabu tenure. The Tabu search, therefore, operates as follows:

1. Perform local search by exploring swap, relocate, and 2opt* neighborhoods.
2. Carry out the best (highest objective function value) move that does not involve any arc on the Tabu list.
3. Mark all arcs involved in the move as Tabu and assign each a Tabu tenure value.

The above three steps are repeated until a stopping condition, such as a time limit, iteration limit, or convergence criterion, is met.

B. *The Case of Insufficient Vehicles for On-Time Delivery of Entire Network*

Due to the sensor network being provided by an external source and the limited availability of delivery resources, it is possible a network cannot be optimally deployed in its entirety. If late delivery of sensors is unacceptable and routes delivering all sensors on-time are not found, one approach that can be taken is to modify the delivery fleet to make on-time delivery of all sensors possible. For example, the vehicle speeds or their numbers could be increased to allow for the

DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

deployment of the complete optimal network.

If the robot fleet could not be modified, an approach of guaranteeing the on-time deployment of the first n' sensors could be taken. This method is, especially, suitable for problems that are dynamic, where the optimal sensor-network configuration could change at any time during the search. It would be, therefore, beneficial to deploy the first n' sensors on-time since they are more likely to be delivered before a change in network configuration occurs.

The above latter approach could be carried out in two phases. Namely, routes would be first planned, during Phase 1 to maximize the minimum spare time for only the first n' sensors, all of which can be deployed on-time. In Phase 2, routes would be planned for the remainder of the network, i.e., $(n - n')$ sensors. Robots would start Phase 2 at their last sensor-deployment locations at the end of Phase 1.

For both abovementioned approaches, a one-dimensional optimization is required to find the minimum number of additional robots required, or the maximum number of sensors that can be deployed on-time, respectively. The one-dimensional optimization can be performed via a simple search algorithm, such as the golden-section search algorithm.

It should be noted that each iteration of the one-dimensional search algorithm would require running our proposed targeted local-search method, to verify whether the number of robots, or sensors, considered, is feasible. Since we propose the use of a metaheuristic optimization method, the proposed algorithm would be run for at most a fixed amount of time every iteration of the one-dimensional search. Thus, the algorithm for addressing the problem of insufficient vehicles has an estimated runtime of the original optimization runtime multiplied by the computational complexity of the one-dimensional optimization algorithm.

For example, if using the golden-section search to determine the maximum number of sensors that can be deployed on-time, the runtime estimate would be equal to the optimization algorithm runtime multiplied by $O(\log(n))$.

If a user decides to forgo the one-dimensional optimization (e.g., due to time constraints) and accept the outcome to simply deploying sensors late, the computational performance of our method would not be impacted. However, by deploying sensors late, this would be equivalent to deploying a sensor network that is suboptimal. Namely, the ability of the sensor network to search for the mobile target would have added uncertainty.

IV. SIMULATED EXPERIMENTS FOR WISAR

Numerous route-planning experiments were performed to validate the effectiveness and robustness of our resource-management method. Two detailed examples are presented in Section IV.A, followed by the presentation and discussion of extensive numerical experiments in Section IV.B. All experiments were performed with the proposed route planning method implemented in MATLAB® 2016a on a Microsoft Windows 10 computer with an Intel® Core™ i7-4770 CPU and 16GB of RAM.

The experiments involved planning routes for a team of robots deploying a static-sensor network, as determined via the method described in [7], in a WiSAR operation, where both sensor deployment times and locations are specified. Thus, maximizing spare time between deployments would improve the ability of the robots to search for the lost target between sensor deliveries.

The sensor network planning method in [7] determines optimal static-sensor deployment times and locations given specific search scenario. The search scenario information at hand is used to predict target motion in the search area, such that sensors can be deployed at locations that maximize the likelihood of target detection. Sensor deployment times are optimized to spread sensor-network's search effort uniformly over time.

Search-scenario information includes the search-area terrain and the characteristics of target motion. Random, realistic terrain information was generated in our work for each scenario using the *Terragen 3* software package [81]. This terrain was, then, used to both compute realistic robot travel times and influence target motion in sensor-network planning. For example, elevation and vegetation information was used to estimate target and robot motion speeds as they moved through the search area. Additionally, large impassable obstacles such as lakes, cliffs, and large boulders were included to impede and direct target and robot movement through the search area.

Target motion characteristics can depend on many factors including age, physical condition, and familiarity with wilderness travel. In [7], the characteristics of target motion were consolidated into three numerical parameters, mean target speed (how fast the target can move), μ, variance of the target's heading (how much the target can *wander*), σ_θ, and the maximum distance for which the target maintains a given heading (how decisive the target is), d_m. For example, a target with a small μ, small σ_θ, and large d_m would be slow, but wander relatively little, and likely to persist in a direction away from their last known position (LKP). In contrast, a target with a larger μ, larger σ_θ, and smaller d_m would be faster, but move more erratically and, thus, likely to propagate outwards at a slower rate. Consequently, the spatial and temporal distribution of sensors in the networks planned for the two different targets would differ. The combined use of realistic terrain to influence target motion and a realistic target motion model results in a sensor network plan that could be deployed for real lost-person search in WiSAR.

Once sensor deployment locations and times are optimized, the method proposed herein is used to determine (near-optimal) robot routes for the (time-phased) deployment of the sensors.

A. Detailed Examples

1) Example 1

In this example, optimal resource management is achieved by planning routes for 15 robots tasked with deploying a network comprising 450 sensors configured to maximize the likelihood of finding a lost target. The target was characterized

DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

by parameters μ = 0.1 m/s, σ_θ = π/4 rad, d_m = 200 m. The network is shown in Fig. 4. While not shown in the figure, each sensor deployment location (blue dot) has a corresponding sensor deployment time.

The robots all begin delivery from a central location, the target's LKP, (red square in Fig. 4) and have identical carrying capacities of 33 sensors each such that the total capacity is 10% more than the network to be deployed (i.e., 45 extra sensors). Robot speeds were randomly defined between 0.7 m/s and 1.3 m/s.

In order to evaluate the effectiveness of our route planning method, we utilized both VNS and TS metaheuristics, where we re-ran each search 30 times (with a time limit of 1800 s) and chose the worst result, respectively, Table I. For comparison purposes, we also obtained (1) a brute-force optimal solution by running the VNS for 12-hours, as well as (2) the best of 1000 random solutions, Table I.

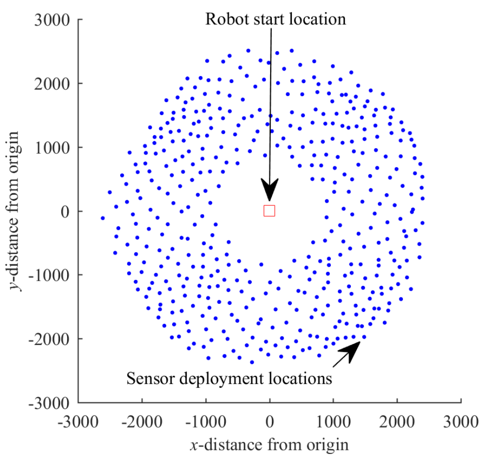


Fig. 4. The sensor network for Example 1.

TABLE I
MINIMUM SPARE TIME VALUES FOR EXAMPLE 1

Worst of 30 VNS (s)	*Worst of 30 TS (s)*	*Brute-force Optimal (s)*	*Best of 1000 Random (s)*
–1468	–2056	–392	–20487

The quality of each solution was assessed based on its proximity to the brute-force (near-global) optimal value normalized with respect to its proximity to the best random value:

$$Q = 100 \times \frac{\tau_{opt} - \tau_{method}}{\tau_{opt} - \tau_{random}} \%, \tag{11}$$

where τ_{method} is the minimum spare time achieved by using our method, τ_{opt} is the minimum spare time of the (near-global) optimal solution, and τ_{random} is the minimum spare time of the best random solution. One may note that a smaller Q value is better, with Q = 0% implying that the method achieved the global optimum spare-time value.

The results obtained show that the proposed method yields routes that are within about 8% of the brute-force (near-global) optimal solution. Plots of the optimization progress for the TS and VNS implementations, respectively, are given in Fig. 5. It can be observed that the quality of our solution (an approximation of the global optimum) improves sub-linearly with increased search time.

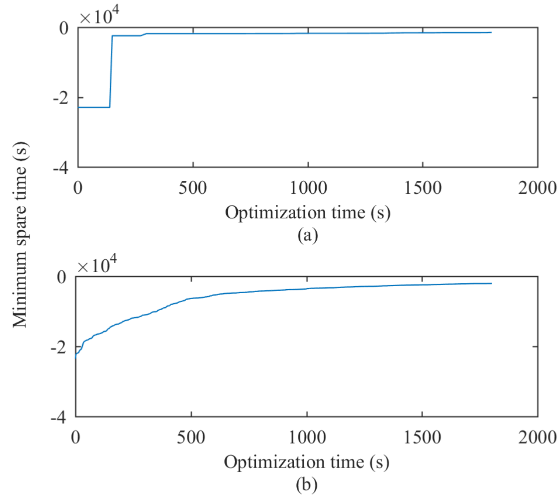


Fig. 5. Optimization progress of the (*a*) VNS and (*b*) TS implementation of the method in solving Example 1.

As noted above, in our example, all 15 robots begin their routes, respectively, at a central location (shown by a red square centered at (0,0) in Fig. 6). They move radially outward, as sensors further away from the origin tend to have later deployment times. The routing solution obtained via VNS is shown in Fig. 6a. The solution obtained via TS is shown in Fig. 6b. Both solutions minimize travel distance between subsequent sensor deployments. However, in some cases the routes backtrack toward the origin. The brute-force optimal solution is shown in Fig. 6c, where the robots remain in their respective sectors and monotonically move outward to maximize minimum spare-time. Lastly, a set of disorganized routes obtained via a random solution is shown in Fig. 6d.

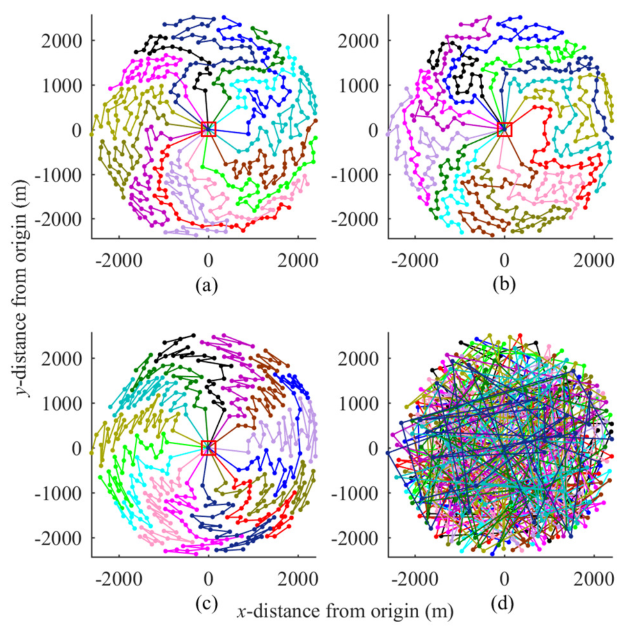


Fig. 6. Routes depicting the solutions obtained for Example 1 from (*a*) worst of solution 30 VNS trials, (*b*) worst of 30 TS trials, (*c*) brute-force optimal, and (*d*) best of 1000 random routes.

DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

It should be noted that, in general, while the visual quality of the solutions can give some indication of the quality of the solution, it is difficult to visually assess the optimality of a set of routes. Namely, since edge values are a combination of spatial and temporal quantities, crossovers in paths do not necessarily imply sub-optimality of the solution.

2) Example 2

This example considers a re-planning situation. Namely, the configuration of the WiSAR sensor network is changed (due to finding a clue regarding target motion) while the robots are in mid-delivery, requiring route re-planning. At the time of re-planning, the 15 robots (red squares) have still need to deploy 150 sensors, and have their delivery routes starting from their most recent locations, Fig. 7.

As in Example 1, while not shown in the figure, each sensor deployment location (blue dot) has a corresponding sensor deployment time. Furthermore, each robot has on-board a different number of sensors available to deploy, in this example ranging from 8 to 14 sensors with a total number of sensors carried being 165 (i.e., there are still 15 redundant sensors distributed amongst the robots). The search area in this example contains a rough terrain, while the target was characterized by parameters μ = 0.12 m/s, $\sigma_\theta = \pi/3$ rad, d_m = 100 m. Robot speeds ranged between 0.99 m/s and 1.48 m/s.

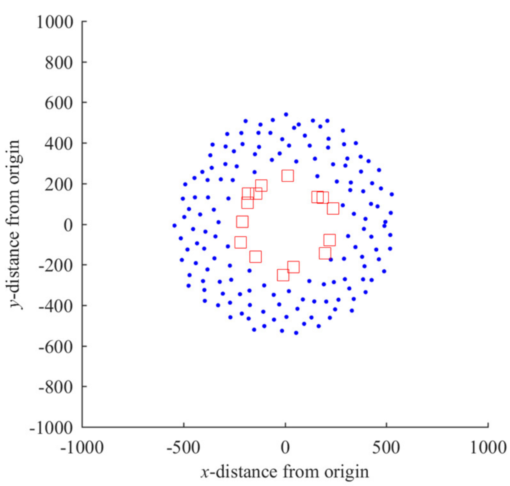


Fig. 7. Illustration of the sensor network being deployed in Example 2.

As in Example 1, the performance of our method in its two implementations was compared to an optimal solution obtained through using an exact method (an integer linear program (ILP) solver) and the best of 1000 random solutions, Table II. In this example, the worst results produced by the method given a 600 s time limit were within <1% of the brute-force optimal solution (as calculated using Eq. (11)).

TABLE II
MINIMUM SPARE TIME VALUES FOR EXAMPLE 2

Worst of 30 VNS (s)	*Worst of 30 TS (s)*	*Brute-force optimal (s)*	*Best of 1000 Random (s)*
274	276	287	–5100

Fig. 8 shows the optimization progress for both implementations of the proposed method. The actual routes for the robots are shown in Fig. 9. As in Fig. 6, robots begin their respective route at a central location and move outwards. The routes in Figs. 9a and 9b are solutions obtained by VNS and TS, respectively. Fig. 9c shows the brute-force optimal solution and Fig. 9d shows the disorganized random solution, respectively.

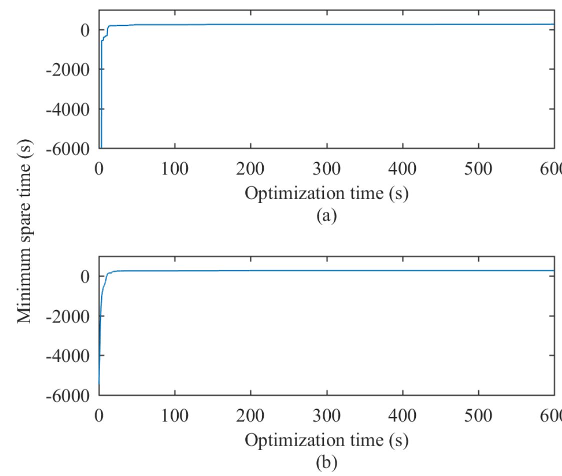


Fig. 8. Optimization progress of the (*a*) VNS and (*b*) TS implementation of the method in solving Example 2.

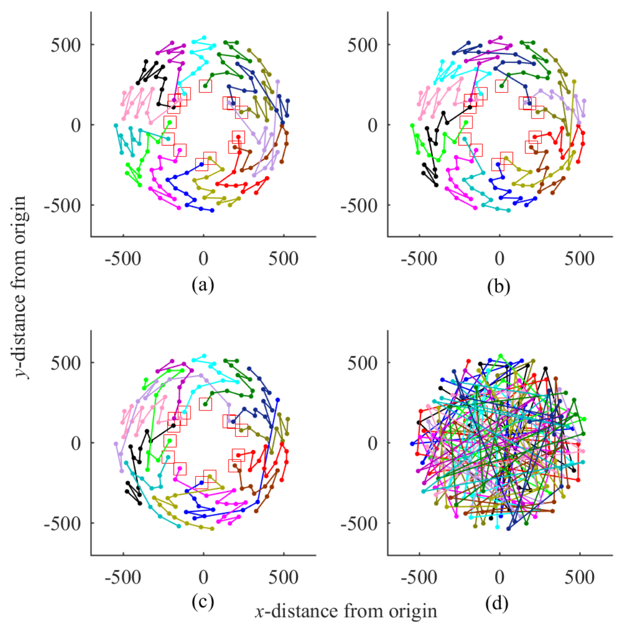


Fig. 9. Routes depicting the solutions obtained for Example 2 from (*a*) worst of solution 30 VNS trials, (*b*) worst of 30 TS trials, (*c*) brute-force optimal, and (*d*) best of 1000 random routes.

B. *Summary of Simulated Experiments*

Twenty-seven distinct combinatorial cases were investigated for each of initial planning and re-planning situations to show the robustness of our proposed method. As indicated in the detailed examples above, initial planning scenarios have all robots starting at the same location with

DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

equal capacities for deployment, while re-planning situations have robots starting at different locations with differing capacities for deployment. The 27 respective cases covered all combinations of three WiSAR scenarios, three network sizes, and three robot fleet sizes, Table III. The three search scenarios are outlined in Table IV.

TABLE III
PARAMETERS FOR EXPERIMENT CASES

	Scenario 1	*Scenario 2*	*Scenario 3*
Network Size	150	300	450
Fleet Size	10	15	20

TABLE IV
PARAMETERS OF THREE SEARCH SCENARIOS

	Scenario 1	*Scenario 2*	*Scenario 3*
Terrain	Flat	Uneven	Uneven
Obstacles	None	Several	None
μ *(m/s)*	0.1	0.12	0.8
σ_θ *(rad)*	$\pi/4$	$\pi/3$	$2\pi/3$
d_m *(m)*	200	100	50

As in the two examples detailed above in Section IV.A, both the VNS and TS implementations of our method were compared to a brute-force (near-global) optimal solution and to the best of 1000 random solutions. Furthermore, the time taken on each problem case to reach within 10% of the brute-force optimal solution was also evaluated to investigate how the performance of the method scales with problem size.

1) Initial Route-Planning Results

The proposed method in both of its implementations was used to solve 27 initial-planning cases, consisting of all combinations of network sizes, fleet sizes, and search scenarios presented in Tables III and IV. Each case was solved for 30 trials. For every trial, the method was allowed to run for 600 s before termination. The quality of each solution obtained was calculated according to Eq. (11), showing the proximity of the solution obtained to the global optimum.

In the 150-sensor network cases, the global optimum was obtained using an exact solution method (an ILP solver). For the larger problems with 300- and 450-sensor networks, however, it was infeasible to obtain exact solutions using an ILP solver. Thus, near-global brute-force optimal solutions were obtained by running VNS for 12 hours.

Despite the variety of search scenario, sensor network size, and fleet size combinations tested, the method was able to consistently produce high quality solutions that maximized the minimum spare time. The results are summarized in Table V. Each entry in the table represents the range of results obtained for each case using our method. No distinction is made between results obtained from the TS or VNS implementation as the two implementations performed similarly.

The primary trend that can be observed from the results presented in Table V is that the quality of the solutions diminishes with increasing sensor-network size. Namely, the proposed method can find an approximation to the global optimal solution that deviate at most 1%, 10%, and 30% from the global optimum objective function value in the 150-, 300-, and 450-sensor network cases, respectively. This indicates that the most significant factor influencing the quality of the solution achieved by our method, given a fixed computational time limit, is the number of sensors in the problem.

TABLE V
SUMMARY OF INITIAL PLANNING TEST RESULTS

Fleet Size	*150 Sensors*	*300 Sensors*	*450 Sensors*
		Scenario 1: Q Values (%)	
10	0–1	3–5	15–23
15	0	2–6	15–29
20	0–1	1–7	18–28
		Scenario 2: Q Values (%)	
10	0	3–6	14–24
15	0	1–6	16–27
20	0	1–4	17–26
		Scenario 3: Q Values (%)	
10	0	0–5	13–23
15	0	0–3	15–27
20	0	0	15–26

Namely, the achieved spare time value is closer to the (global) optimal value, if there are fewer sensors to route. This is due to a larger number of sensors increasing computation time and slowing the search engine, as well as the search space increasing in size. Variations in solution quality across fleet size and scenarios, however, are comparatively small. This suggests that the sensor-network configuration, the spare-time values, and the number of robots have comparatively minor impacts on the method's ability to optimize routes.

Further investigation of how the search scenario, sensor network size, and fleet size influence the method's performance was carried out by investigating the time required to achieve 10% proximity to the optimal solution. Fig. 10 shows the results of the experiments. Each line in the graph indicates a different fleet size as indicated by the legend. Data points indicate the latest time of achieving 10% for a given network and fleet size aggregated over the two implementations of the method and search scenarios. Aggregation was performed over the implementation and search scenarios as there was no significant difference in performance over these parameters. This indicates that our proposed method performs similarly among different spatial and temporal distributions of the sensor networks.

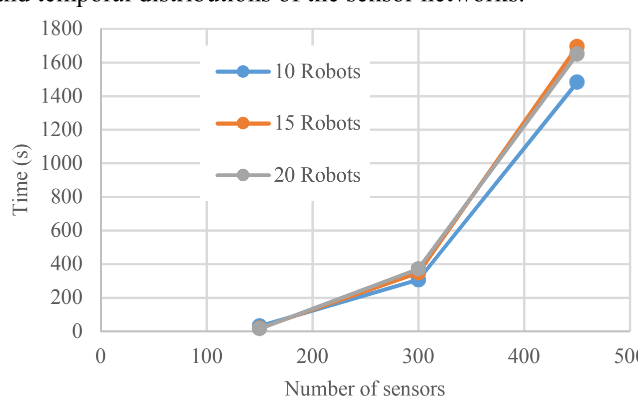


Fig. 10. Time to reach within 10% the optimal solution for various cases of initial planning.

DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

From Fig. 10, it can also be noted that the number of sensors influences the time required to achieve a given quality solution more than the fleet size. Furthermore, the results indicate that the time required scales super-linearly with the number of sensors.

2) Route Re-Planning Results

Both implementations of the proposed method were also used to obtain solutions to 27 re-planning cases, consisting of all combinations of network sizes, fleet sizes, and search scenarios presented in Tables III and IV. Each case was solved for 30 trials with a time limit of 600 s. On may recall that re-planning cases have robots starting at unique locations with different capacities for deployment as shown in the second detailed example. The solution quality values were again calculated according to Eq. (11) to show solution proximity to the global optimum.

As in the initial planning cases, the global optimum for the 150-sensor network cases was obtained using an ILP solver. Similarly, near-global brute-force optimal solutions were obtained by running VNS for 12 hours for the 300- and 450-sensor network cases. A summary of the worst results obtained from using the proposed method is presented in Table VI.

TABLE VI
SUMMARY OF RE-PLANNING TEST RESULTS (%)

Fleet Size	*150 Sensors*	*300 Sensors*	*450 Sensors*
	Scenario 1: Q Values (%)		
10	0–1	3–5	16–23
15	0–1	5–6	14–27
20	1–2	2–9	17–24
	Scenario 2: Q Values (%)		
10	4–5	3–6	15–24
15	0	1–7	16–27
20	0	1–8	16–26
	Scenario 3: Q Values (%)		
10	0–1	0–5	17–22
15	0	0–6	15–22
20	0	0–1	15–23

As with the initial planning case, the results suggest that the primary influence on the quality of solution is the number of sensors in the network being deployed. Similarly, variations in quality across fleet sizes and scenarios is small as in the initial planning case. This further indicates that the network configuration, spare-time values, and number of robots have minor impacts on route-planning performance. Furthermore, comparing the results presented in Table V with those in Table VI, it is possible to note that whether the problem is one of initial planning *vs* re-planning has also comparatively little impact on route-planning performance.

Overall, the results indicate that the proposed method is capable of effectively optimizing routes given both initial planning and re-planning situations in addition to various sensor numbers, fleet sizes, and search scenarios.

Additional investigation revealed that the worst-case time required by the method to achieve a value within 10% of the optimal varies is again super-linearly related to the number of sensors in the network, Fig. 11.

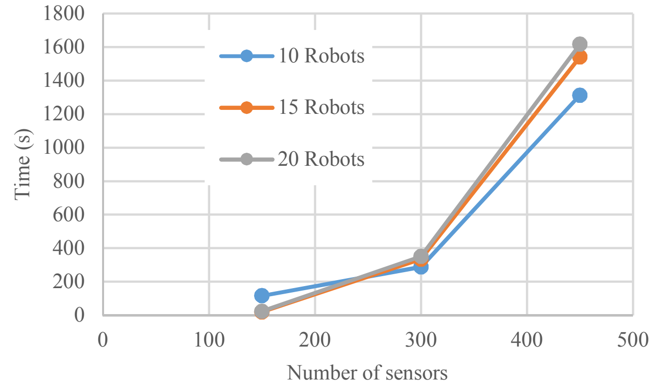


Fig. 11. Time to reach within 10% the optimal solution for various cases of re-planning.

C. Comparisons

We compared several different approaches to route optimization including Simulated Annealing, Ant Colony Optimization, Genetic Algorithm, and Variable Neighborhood Search (VNS) and Tabu Search (TS). The comparisons revealed that the local-search based VNS and TS yielded the best results. The comparative results are shown in Table VII for initial planning, and Table VIII for re-planning. As can be noted from the tables, best Q results were obtained using VNS/TS with targeted local search.

TABLE VII
COMPARISON OF OPTIMIZATION METHODOLOGIES FOR INITIAL PLANNING

# Sensors, # Robots, Scenario #	*Best of SA (%)*	*Best of ACO (%)*	*Best of GA (%)*	*Best of Proposed VNS/TS (%)*	*Worst of Proposed VNS/TS (%)*
150, 10, 3	1	56	5	≈0	≈0
150, 20, 2	1	62	12	≈0	≈0
300, 10, 1	28	64	17	1	4
300, 15, 2	30	67	12	≈0	1
450, 15, 3	49	73	4	≈0	9

TABLE VIII
COMPARISON OF OPTIMIZATION METHODOLOGIES FOR RE-PLANNING

# Sensors, # Robots, Scenario #	*Best of SA (%)*	*Best of ACO (%)*	*Best of GA (%)*	*Best of Proposed VNS/TS (%)*	*Worst of Proposed VNS/TS (%)*
150, 15, 1	11	63	18	≈0	1
300, 20, 3	34	68	6	≈0	1
450, 10, 2	43	69	11	2	6
450, 20, 1	52	77	21	4	8

Additional simulations were run comparing the proposed method to a greedy algorithm for optimizing routes. In the greedy algorithm, sensors were assigned to robots sequentially, starting with Sensor 1 and continuing until Sensor n. The assignment that results in the maximum spare time is selected at every step. The results indicated that our

DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

proposed VNS/TS method consistently outperforms the greedy algorithm for reasonable-sized problem, e.g., the 150-sensor/10-robot and the 300-sensor/15-robot cases. However, the greedy algorithm would outperform our method for large-sized problems, e.g., the 450-sensor/20-robot case, unless the latter is given significantly more computation time, which would not be realistic for most applications.

V. Conclusions

In this paper, we address the resource-management problem for time-phased sensor-network deployments, for applications such as wilderness search and rescue and wildfire monitoring. In these applications, it would be beneficial to maximize the *spare time* available to the delivery vehicles, between scheduled deployments, such that they can perform other tasks. Since the sensors need to be deployed to specific locations at predetermined corresponding times, the problem at hand is unique and necessitates a novel vehicle route-planning formulation that includes our spare-time objective function. Furthermore, achieving an unbiased distribution of spare time over the course of network deployment requires the use of an effective objective function, in our case, maximizing minimum spare time between sensor deliveries.

In order to address this problem, we developed a method that includes a novel targeted local search algorithm for route planning. The proposed algorithm is novel in that it targets a single arc in the solution to improve upon, narrowing the search space significantly. This algorithm can be used as a basis for metaheuristic solution methods, as has been demonstrated by its incorporation into the frameworks of both variable neighborhood search and Tabu search.

Experiments in the context of a WiSAR operation were presented to validate the proposed method, to illustrate its efficiency, as well as its robustness to a variety of parameter values and conditions. The obtained (worst-case) solutions were compared to brute-force and (best-case) random solutions.

Although demonstrated in the context of WiSAR, the proposed method can be adapted to various time-constrained route-planning problems.

A topic of interest that can be considered for future work is having robots with limited battery life. This could be addressed in several ways, including incorporating re-charging trips to a re-charging station during delivery or having other robots rendezvous with the delivery robots to offer re-charging. Limited sensor carrying capacity could also be addressed in a similar manner.

References

[1] I. Martin *et al.*, "A High-Resolution Sensor Network for Monitoring Glacier Dynamics," *IEEE Sensors J.*, vol. 14, no. 11, pp. 3926–3931, Nov. 2014.
[2] A. Bakhtari, M. D. Naish, M. Eskandari, E. A. Croft, and B. Benhabib, "Active-vision-based multisensor surveillance - an implementation," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 36, no. 5, pp. 668–680, Sep. 2006.
[3] R. Lara, D. Benitez, A. Caamano, M. Zennaro, and J. L. Rojo-Alvarez, "On Real-Time Performance Evaluation of Volcano-Monitoring Systems with Wireless Sensor Networks," *IEEE Sensors J.*, vol. 15, no. 6, pp. 3514–3523, Jun. 2015.
[4] X. Xie, J. Guo, H. Zhang, T. Jiang, R. Bie, and Y. Sun, "Neural-Network Based Structural Health Monitoring with Wireless Sensor Networks," in *Proc. Int. Conf. Natural Computation,* Shenyang, China, 2013, pp. 163–167.
[5] J. Vilela, Z. Kashino, R. Ly, G. Nejat, and B. Benhabib, "A Dynamic Approach to Sensor Network Deployment for Mobile-Target Detection in Unstructured, Expanding Search Areas," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4405–4417, Jun. 2016.
[6] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*. John Wiley & Sons, 2010.
[7] Z. Kashino, J. Y. Kim, G. Nejat, and B. Benhabib, "Spatiotemporal Adaptive Optimization of a Static-Sensor Network via a Non-Parametric Estimation of Target Location Likelihood," *IEEE Sensors J.*, vol. 17, no. 5, pp. 1479–1492, Mar. 2017.
[8] T. Arampatzis, J. Lygeros, and S. Manesis, "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks," in *Proc. IEEE Int. Symp. Mediterrean Conf. Control Automation Intell. Control*, Limassol, Cyprus, 2005, pp. 719–724.
[9] Z. Zhang, G. Nejat, H. Guo, and P. Huang, "A novel 3D sensory system for robot-assisted mapping of cluttered urban search and rescue environments," *Intell. Serv. Robot.*, vol. 4, no. 2, pp. 119–134, Apr. 2011.
[10] B. R. Le Comte, G. Sen Gupta, and M. T. Chew, "Distributed sensors for hazard detection in an urban search and rescue operation," in *IEEE Int. Instrumentation and Measurement Technology Conf.,* Graz, Austria, 2012, pp. 2385–2390.
[11] B. Doroodgar, Y. Liu, and G. Nejat, "A Learning-Based Semi-Autonomous Controller for Robotic Exploration of Unknown Disaster Scenes While Searching for Victims," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2719–2732, Dec. 2014.
[12] M. D. Mackay, R. G. Fenton, and B. Benhabib, "Multi-camera active surveillance of an articulated human form – An implementation strategy," *Comput. Vis. Image Underst.*, vol. 115, no. 10, pp. 1395–1413, Oct. 2011.
[13] R. Bellazreg, N. Boudriga, and S. An, "Border Surveillance Using Sensor Based Thick-Lines," in *Proc. Int. Conf. Inform. Netw.,* Bangkok, Thailand, 2013, pp. 221–226.
[14] A. Bakhtari, M. Mackay, and B. Benhabib, "Active-Vision for the Autonomous Surveillance of Dynamic, Multi-Object Environments," *J. Intell. Robot. Syst.*, vol. 54, no. 4, p. 567, Apr. 2009.
[15] A. Arora *et al.*, "A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking," *Comput. Netw.*, vol. 46, no. 5, pp. 605–634, Dec. 2004.
[16] Y. Liu and G. Nejat, "Multirobot Cooperative Learning for Semiautonomous Control in Urban Search and Rescue Applications," *J. Field Robot.*, vol. 33, no. 4, pp. 512–536, Jun. 2016.
[17] A. Bakhtari and B. Benhabib, "An Active Vision System for Multitarget Surveillance in Dynamic Environments," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 37, no. 1, pp. 190–198, Feb. 2007.
[18] Z. Bojkovic and B. Bakmaz, "A Survey on Wireless Sensor Networks Deployment," *WSEAS Trans. Commun.*, vol. 7, no. 12, pp. 1172–1181, Dec. 2008.
[19] D. S. Deif and Y. Gadallah, "Classification of Wireless Sensor Networks Deployment Techniques," *IEEE Commun. Surveys Tutorials*, vol. 16, no. 2, pp. 834–855, Second 2014.
[20] M. Xiang, L. Sun, and L. Li, "Survey on the Connectivity and Coverage in Wireless Sensor Networks," in *Proc. Int. Conf. Wireless Commun. Netw. Mobile Comput.,* Wuhan, China, 2011, pp. 1–4.
[21] S. Mini, S. K. Udgata, and S. L. Sabat, "Sensor Deployment and Scheduling for Target Coverage Problem in Wireless Sensor Networks," *IEEE Sensors J.*, vol. 14, no. 3, pp. 636–644, Mar. 2014.
[22] J. Du, K. Wang, H. Liu, and D. Guo, "Maximizing the Lifetime of k-Discrete Barrier Coverage Using Mobile Sensors," *IEEE Sensors J.*, vol. 13, no. 12, pp. 4690–4701, Dec. 2013.
[23] M. R. Ingle and N. Bawane, "An energy efficient deployment of nodes in wireless sensor network using Voronoi diagram," in *Int. Conf. Electron. Comput. Tech.*, Kanyakumari, India, 2011, vol. 6, pp. 307–311.
[24] A. Macwan, G. Nejat, and B. Benhabib, "Optimal deployment of robotic teams for autonomous wilderness search and rescue," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisto, USA, 2011, pp. 4544–4549.

DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

[25] A. Macwan, J. Vilela, G. Nejat, and B. Benhabib, “A Multirobot Path-Planning Strategy for Autonomous Wilderness Search and Rescue,” *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1784–1797, Sep. 2015.

[26] A. Macwan, G. Nejat, and B. Benhabib, “Target-Motion Prediction for Robotic Search and Rescue in Wilderness Environments,” *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 41, no. 5, pp. 1287–1298, Oct. 2011.

[27] D. W. Casbeer, R. W. Beard, T. W. McLain, S.-M. Li, and R. K. Mehra, “Forest Fire Monitoring with Multiple Small UAVs,” in *Proc. American Control Conf.,* Portland, USA, *2005*., 2005, pp. 3530–3535 vol. 5.

[28] A. Belbachir, J. Escareno, E. Rubio, and H. Sossa, “Preliminary Results on UAV-Based Forest Fire Localization Based on Decisional Navigation,” in *Workshop Research Educ. Develop. Unmanned Aerial Syst.,* Cancun, Mexico, 2015, pp. 377–382.

[29] K. Sundar and S. Rathinam, “Algorithms for Routing an Unmanned Aerial Vehicle in the Presence of Refueling Depots,” *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 287–294, Jan. 2014.

[30] P. Pounds and S. Singh, “Samara: Biologically Inspired Self-Deploying Sensor Networks,” *IEEE Potentials*, vol. 34, no. 2, pp. 10–14, Mar. 2015.

[31] G. Laporte, “Fifty Years of Vehicle Routing,” *Transp. Sci.*, vol. 43, no. 4, pp. 408–416, Nov. 2009.

[32] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, “Vehicle Routing Problems for Drone Delivery,” *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017.

[33] N. Mathew, S. L. Smith, and S. L. Waslander, “Planning Paths for Package Delivery in Heterogeneous Multirobot Teams,” *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1298–1308, Oct. 2015.

[34] L. Abbatecola, M. P. Fanti, A. M. Mangini, and W. Ukovich, “A Decision Support Approach for Postal Delivery and Waste Collection Services,” *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 4, pp. 1458–1470, Oct. 2016.

[35] Y. Han, X. Guan, and L. Shi, “Optimization Based Method for Supply Location Selection and Routing in Large-Scale Emergency Material Delivery,” *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 4, pp. 683–693, Oct. 2011.

[36] E. K. Hashi, M. R. Hasan, and M. S. U. Zaman, “GIS Based Heuristic Solution of the Vehicle Routing Problem to Optimize the School Bus Routing and Scheduling,” in *Proc. Int. Conf. Comput. Inform. Tech.,* Dhaka, Bangladesh, 2016, pp. 56–60.

[37] S. Souyris, C. E. Cortés, F. Ordóñez, and A. Weintraub, “A Robust Optimization Approach to Dispatching Technicians Under Stochastic Service Times,” *Optim. Lett.*, vol. 7, no. 7, pp. 1549–1568, Oct. 2013.

[38] R. Falcon, X. Li, A. Nayak, and I. Stojmenovic, “A Harmony-Seeking Firefly Swarm to the Periodic Replacement of Damaged Sensors by a Team of Mobile Robots,” in *Proc. IEEE Int. Conf. Commun.,* Ottawa, Canada, 2012, pp. 4914–4918.

[39] D. Kim, R. N. Uma, B. H. Abay, W. Wu, W. Wang, and A. O. Tokuta, “Minimum Latency Multiple Data MULE Trajectory Planning in Wireless Sensor Networks,” *IEEE Trans. Mob. Comput.*, vol. 13, no. 4, pp. 838–851, Apr. 2014.

[40] Y. Wang and C. H. Wu, “Robot-Assisted Sensor Network Deployment and Data Collection,” in *Proc. Int. Symp. Comput. Intell. Robotics Automation,* Jacksonville, USA, 2007, pp. 467–472.

[41] M. A. Mohammed *et al.*, “Solving Vehicle Routing Problem by Using Improved K-Nearest Neighbor Algorithm for Best Solution,” *J. Comput. Sci.*, vol. 21, pp. 232–240, Jul. 2017.

[42] A. A. R. Hosseinabadi, N. S. H. Rostami, M. Kardgar, S. Mirkamali, and A. Abraham, “A New Efficient Approach for Solving the Capacitated Vehicle Routing Problem Using the Gravitational Emulation Local Search Algorithm,” *Appl. Math. Model.*, vol. 49, pp. 663–679, Sep. 2017.

[43] E. Onieva, E. Osaba, I. Angulo, A. Moreno, A. Bahillo, and A. Perallos, “Improvement of Drug Delivery Routes Through the Adoption of Multi-Operator Evolutionary Algorithms and Intelligent Vans Capable of Reporting Real-Time Incidents,” *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1009–1019, Apr. 2017.

[44] F. Li, B. Golden, and E. Wasil, “The Open Vehicle Routing Problem: Algorithms, Large-Scale Test Problems, and Computational Results,” *Comput. Oper. Res.*, vol. 34, no. 10, pp. 2918–2930, Oct. 2007.

[45] Z. Özyurt, D. Aksen, and N. Aras, “Open Vehicle Routing Problem with Time Deadlines: Solution Methods and an Application,” in *Operations Research Proc.*, Springer, Berlin, Heidelberg, 2006, pp. 73–78.

[46] J. R. Montoya-Torres, J. López Franco, S. Nieto Isaza, H. Felizzola Jiménez, and N. Herazo-Padilla, “A Literature Review on the Vehicle Routing Problem with Multiple Depots,” *Comput. Ind. Eng.*, vol. 79, pp. 115–129, Jan. 2015.

[47] J. Jiang, K. M. Ng, K. L. Poh, and K. M. Teo, “Vehicle Routing Problem with a Heterogeneous Fleet and Time Windows,” *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3748–3760, Jun. 2014.

[48] P. H. V. Penna, A. Subramanian, and L. S. Ochi, “An Iterated Local Search heuristic for the Heterogeneous Fleet Vehicle Routing Problem,” *J. Heuristics*, vol. 19, no. 2, pp. 201–232, Apr. 2013.

[49] O. Bräysy and M. Gendreau, “Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms,” *Transp. Sci.*, vol. 39, no. 1, pp. 104–118, Feb. 2005.

[50] O. Bräysy and M. Gendreau, “Vehicle Routing Problem with Time Windows, Part II: Metaheuristics,” *Transp. Sci.*, vol. 39, no. 1, pp. 119–139, Feb. 2005.

[51] N. Smolic-Rocak, S. Bogdan, Z. Kovacic, and T. Petrovic, “Time Windows Based Dynamic Routing in Multi-AGV Systems,” *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 151–155, Jan. 2010.

[52] S. R. Thangiah, I. H. Osman, R. Vinayagamoorthy, and T. Sun, “Algorithms for the Vehicle Routing Problems with Time Deadlines,” *American J. Math. Manage. Sci.*, vol. 13, no. 3–4, pp. 323–355, Apr. 1993.

[53] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, “A Review of Dynamic Vehicle Routing Problems,” *Eur. J. Oper. Res.*, vol. 225, no. 1, pp. 1–11, Feb. 2013.

[54] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, “Heuristics for Multi-Attribute Vehicle Routing Problems: a Survey and Synthesis,” *Eur. J. Oper. Res.*, vol. 231, no. 1, pp. 1–21, Nov. 2013.

[55] J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, and A. A. Juan, “Rich Vehicle Routing Problem: Survey,” *ACM Comput. Surv.*, vol. 47, no. 2, pp. 1–28, Dec. 2014.

[56] N. Jozefowiez, F. Semet, and E.-G. Talbi, “Multi-Objective Vehicle Routing Problems,” *Eur. J. Oper. Res.*, vol. 189, no. 2, pp. 293–309, Sep. 2008.

[57] S. U. Ngueveu, C. Prins, and R. Wolfler Calvo, “An Effective Memetic Algorithm for the Cumulative Capacitated Vehicle Routing Problem,” *Comput. Oper. Res.*, vol. 37, no. 11, pp. 1877–1885, Nov. 2010.

[58] M. Barkaoui, J. Berger, and A. Boukhtouta, “Customer Satisfaction in Dynamic Vehicle Routing Problem with Time Windows,” *Appl. Soft Comput.*, vol. 35, pp. 423–432, Oct. 2015.

[59] Tzong-Ru Lee and Ji-Hwa Ueng, “A Study of Vehicle Routing Problems with Load-Balancing,” *Int. J. Phys. Distrib. Logist. Manag.*, vol. 29, no. 10, pp. 646–657, Dec. 1999.

[60] D. Applegate, W. Cook, S. Dash, and A. Rohe, “Solution of a Min-Max Vehicle Routing Problem,” *Inf. J. Comput.*, vol. 14, no. 2, pp. 132–143, May 2002.

[61] J. LaRusic and A. P. Punnen, “The Asymmetric Bottleneck Traveling Salesman Problem: Algorithms, Complexity and Empirical Analysis,” *Comput. Oper. Res.*, vol. 43, pp. 20–35, Mar. 2014.

[62] J.-F. Cordeau and G. Laporte, “Tabu Search Heuristics for the Vehicle Routing Problem,” in *Metaheuristic Optimization via Memory and Evolution*, Springer, Boston, MA, 2005, pp. 145–163.

[63] F. Glover and M. Laguna, *Tabu Search*. Norwell, MA, USA: Kluwer Academic Publishers, 1997.

[64] P. Toth and D. Vigo, “The Granular Tabu Search and Its Application to the Vehicle-Routing Problem,” *Inf. J Comput.*, vol. 15, no. 4, pp. 333–346, Dec. 2003.

[65] J.-F. Cordeau and M. Maischberger, “A Parallel Iterated Tabu Search Heuristic for Vehicle Routing Problems,” *Comput. Oper. Res.*, vol. 39, no. 9, pp. 2033–2050, Sep. 2012.

[66] P. Hansen and N. Mladenović, “An Introduction to Variable Neighborhood Search,” in *Meta-Heuristics*, Springer, Boston, MA, 1999, pp. 433–458.

[67] J. Kytöjoki, T. Nuortio, O. Bräysy, and M. Gendreau, “An Efficient Variable Neighborhood Search Heuristic for Very Large Scale Vehicle Routing Problems,” *Comput. Oper. Res.*, vol. 34, no. 9, pp. 2743–2757, Sep. 2007.

[68] J. de Armas and B. Melián-Batista, “Variable Neighborhood Search for a Dynamic Rich Vehicle Routing Problem with time windows,” *Comput. Ind. Eng.*, vol. 85, pp. 120–131, Jul. 2015.

[69] S. Afifi, D.-C. Dang, and A. Moukrim, “A Simulated Annealing Algorithm for the Vehicle Routing Problem with Time Windows and Synchronization Constraints,” in *Learning and Intelligent*

This is the author's version of an article that has been published in IEEE Transactions on Automation Science and Engineering.
DOI: 10.1109/TASE.2018.2857630
The final version of the paper is available at https://ieeexplore.ieee.org/document/8425715

Optimization, vol. 7997, G. Nicosia and P. Pardalos, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 259–265.

[70] V. F. Yu, A. A. N. P. Redi, Y. A. Hidayat, and O. J. Wibowo, "A Simulated Annealing Heuristic for the Hybrid Vehicle Routing Problem," *Appl. Soft Comput.*, vol. 53, pp. 119–132, Apr. 2017.

[71] B. Bullnheimer, R. F. Hartl, and C. Strauss, "An Improved Ant System Algorithm for the Vehicle Routing Problem," *Annals Oper. Res.*, vol. 89, no. 0, pp. 319–328, Jan. 1999.

[72] K. Venkata Narasimha, E. Kivelevitch, B. Sharma, and M. Kumar, "An Ant Colony Optimization Technique for Solving Min–Max Multi-Depot Vehicle Routing Problem," *Swarm Evol. Comput.*, vol. 13, pp. 63–73, Dec. 2013.

[73] J.-Y. Potvin, "State-of-the Art Review—Evolutionary Algorithms for Vehicle Routing," *Inf. J. Comput.*, vol. 21, no. 4, pp. 518–548, Apr. 2009.

[74] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A Hybrid Genetic Algorithm with Adaptive Diversity Management for a Large Class of Vehicle Routing Problems with Time-Windows," *Comput. Oper. Res.*, vol. 40, no. 1, pp. 475–489, Jan. 2013.

[75] Y. Nagata and O. Bräysy, "Edge Assembly-Based Memetic Algorithm for the Capacitated Vehicle Routing Problem," *Networks*, vol. 54, no. 4, pp. 205–215, Dec. 2009.

[76] H. C. W. Lau, T. M. Chan, W. T. Tsui, and W. K. Pang, "Application of Genetic Algorithms to Solve the Multidepot Vehicle Routing Problem," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 2, pp. 383–392, Apr. 2010.

[77] M. A. Mohammed, M. K. Abd Ghani, R. I. Hamed, S. A. Mostafa, M. S. Ahmad, and D. A. Ibrahim, "Solving Vehicle Routing Problem by Using Improved Genetic Algorithm for Optimal Solution," *J. Comput. Sci.*, vol. 21, pp. 255–262, Jul. 2017.

[78] A. Chen, G. Yang, and Z. Wu, "Hybrid Discrete Particle Swarm Optimization Algorithm for Capacitated Vehicle Routing Problem," *J. Zhejiang Univ.-Sci. A*, vol. 7, no. 4, pp. 607–614, Apr. 2006.

[79] K. F. Doerner, R. F. Hartl, "Health Care Logistics, Emergency Preparedness, and Disaster Relief: New Challenges for Routing Problems with a Focus on the Austrian Situation," in *The Vehicle Routing Problem: Latest Advances and New Challenges*, London: Springer, 2008.

[80] C. Wang, M. Dessouky, and F. Ordóñez, "Routing Courier Delivery Services with Urgent Demand," *INFOR Inf. Syst. Oper. Res.*, vol. 53, no. 1, pp. 26–39, Feb. 2015.

[81] "Terragen 3." [Online]. Available: http://planetside.co.uk/products/terragen3. [Accessed: 25-Feb-2016].

Kasper Woiceshyn received his B.A.Sc. degree in engineering science in 2012 and his M.Eng. degree in mechanical engineering in 2017, both from the University of Toronto. His graduate research work focused on addressing the route planning problem for sensor delivery in sensory network applications.

Zendai Kashino (S'16) received the B.A.Sc. degree in engineering physics from the University of British Columbia, Vancouver, BC, Canada, in 2015. He is currently pursuing the Ph.D. degree in mechanical engineering at the University of Toronto under the supervision of Prof. Benhabib and Prof. Nejat, with a focus on autonomous wilderness search and rescue.

Goldie Nejat (S'03–M'06) received the B.A.Sc. and Ph.D. degrees in mechanical engineering from UofT in 2001 and 2005, respectively. She is currently the Canada Research Chair in Robots for Society, and also an Associate Professor and the Director of the Autonomous Systems and Biomechatronics Lab in the Department of Mechanical and Industrial Engineering, UofT. Her research interests include sensing, HRI, semi-autonomous and autonomous control, and intelligence of assistive/service robots for search and rescue, exploration, healthcare, and surveillance applications.

Beno Benhabib received the B.Sc., M.Sc., and Ph.D. degrees all in mechanical engineering in 1980, 1982, and 1985, respectively. He has been a Professor with the Department of Mechanical and Industrial Engineering, and the Institute for Biomaterials and Biomedical Engineering at the University of Toronto, Toronto, ON, Canada, since 1986. His current research interests include the design and control of intelligent autonomous systems.

Appendix A

The algorithm for each type of Local Search (LS) is presented below.

Algorithm A1: Targeted Swap LS

```
while bestImprovement > 0
    bestImprovement = 0
    find lowest arc value, minVal, identify surrounding nodes B and C
    for every node Y in other routes
        if f_swap(B, Y) – minVal > bestImprovement
            bestImprovement = f_swap(B, Y) – minVal
            record swap(B, Y) as bestMove
        if f_swap(C, Y) – minVal > bestImprovement
            bestImprovement = f_swap(C, Y) – minVal
            record swap(C, Y) as bestMove
    if bestImprovement > 0
        execute bestMove
```

Algorithm A2: Targeted Relocate LS

```
while bestImprovement > 0
    bestImprovement = 0
    find lowest arc value, minVal, identify surrounding nodes B and C
    for every node Y in other routes
        if f_relocate(B, Y) – minVal > bestImprovement
            bestImprovement = f_relocate(B, Y) – minVal
            record relocate(B, Y) as bestMove
        if f_relocate(C, Y) – minVal > bestImprovement
            bestImprovement = f_relocate(C, Y) – minVal
            record relocate(C, Y) as bestMove
        if f_relocate(Y, B) – minVal > bestImprovement
            bestImprovement = f_relocate(Y, B) – minVal
            record relocate(Y, B) as bestMove
    if bestImprovement > 0
        execute bestMove
```

Algorithm A3: Targeted 2OPT* LS

```
while bestImprovement > 0
    bestImprovement = 0
    find lowest arc value, minVal, identify predecessor node B
    for every node Y in other routes
        if f_2opt*(B, Y) – minVal > bestImprovement
            bestImprovement = f_2opt*(B, Y) – minVal
            record 2opt*(B, Y) as bestMove
    if bestImprovement > 0
        execute bestMove
```