Journal of Intelligent & Robotic Systems (2024) 110:134
https://doi.org/10.1007/s10846-024-02169-1

REGULAR PAPER

UAV Routing for Enhancing the Performance of a Classifier-in-the-loop

Deepak Prakash Kumar[1] · Pranav Rajbhandari[2] · Loy McGuire[3] · Swaroop Darbha[1] · Donald Sofge[4]

Received: 24 February 2024 / Accepted: 22 August 2024 / Published online: 14 September 2024

Abstract
Some human-machine systems are designed so that machines (robots) gather and deliver data to remotely located operators (humans) through an interface to aid them in classification. The performance of a human as a (binary) classifier-in-the-loop is characterized by probabilities of correctly classifying objects (or points of interest) as a true target or a false target. These two probabilities depend on the time spent collecting information at a point of interest (POI), known as dwell time. The information gain associated with collecting information at a POI is then a function of dwell time and discounted by the revisit time, i.e., the duration between consecutive revisits to the same POI, to ensure that the vehicle covers all POIs in a timely manner. The objective of the routing problem for classification is to route the vehicles optimally, which is a discrete problem, and determine the optimal dwell time at each POI, which is a continuous optimization problem, to maximize the total discounted information gain while visiting every POI at least once. Due to the coupled discrete and continuous problem, which makes the problem hard to solve, we make a simplifying assumption that the information gain is discounted exponentially by the revisit time; this assumption enables one to decouple the problem of routing with the problem of determining optimal dwell time at each POI for a single vehicle problem. For the multi-vehicle problem, since the problem involves task partitioning between vehicles in addition to routing and dwell time computation, we provide a fast heuristic to obtain high-quality feasible solutions.

Keywords Aerial systems · Applications · Heuristic · Routing · Task allocation · Operator-in-the-loop

✉ Deepak Prakash Kumar
deepakprakash1997@gmail.com

Pranav Rajbhandari
prajbhan@alumni.cmu.edu

Loy McGuire
loy.j.mcguire.civ@us.navy.mil

Swaroop Darbha
dswaroop@tamu.edu

Donald Sofge
donald.a.sofge.civ@us.navy.mil

[1] Department of Mechanical Engineering, Texas A&M University, 3123 TAMU, College Station, TX 77843, USA

[2] Department of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

[3] Department of Aerospace Engineering, University of Maryland, 4298 Campus Drive, College Park, MD 20742, USA

[4] Distributed Autonomous Systems, Navy Center for Applied Research in Artificial Intelligence, US Naval Research Laboratory, 4555 Overlook Avenue SW, Washington, DC 20375, USA

1 Introduction

The design of human-machine systems requires a careful partition of the tasks to be performed by the human-in-the-loop and the machines, as well as the design of an associated human-machine interface. In this paper, we consider a human-machine system, where the human serves as a classifier-in-the-loop based on the information delivered to the human operator by the machines (vehicles) through an interface. Through the coupling of the human and machine, we seek to utilize the ability of machines/vehicles, such as unmanned aerial vehicles (UAVs), to reach remote locations and obtain quality information, such as images and videos, back to the classifier/human to make informed decisions. Such a design is applicable in surveillance applications, wherein the human identifies n suspicious areas of activity, denoted as Points of Interest (POI) or targets, and provides them to the interface. The interface additionally takes as input the number, m, of UAVs used for classification and computes

- the allocation of targets to each vehicle,

- the order in which the allocated targets must be visited by each vehicle, and
- the time to be spent at each POI to collect information such as images/videos.

The vehicles persistently monitor the targets by visiting them, i.e., visiting the targets periodically and dwelling at a POI while collecting information that is transmitted to the remotely located operator. Based on the information supplied by the vehicles, the primary task of the human is to assess/classify events happening at the n specified targets and classify them as T (true target) or F (false target). An overview of the described human-machine system is provided in Fig. 1.

The probability of an operator correctly classifying events at a target depends on the dwell time of the vehicle at that location and the revisit time, R, i.e., the time duration between consecutive revisits to the same location. The dependence on the dwell time is expressed through the Kullback-Leibler divergence, which is widely used to measure similarities or differences between two distributions [1] and, hence, used for classification. Due to this reason, Kullback-Leibler divergence, also denoted as mutual information gain, has also been used in the literature for the calibration of sensors as well, wherein the goal is to obtain calibration parameters that maximize the information gain [2]. In our study, we model a discounted information gain to reflect the trade-off between the information gained as a function of dwell time d and the revisit time, R, without which the vehicle will remain at a particular POI indefinitely. Since the problem involves determining the allocation of targets to each vehicle and the route that each vehicle needs to take, both of which are decisions involving discrete variables, and the computation of the dwell time at each target, which is a continuous variable, we first consider a single-vehicle problem. In the single-vehicle problem, all targets are covered by a single vehicle, reducing the problem to computing the vehicle route and dwell time. Under a simplifying assumption that the information gain depends on R through an exponential function and the edge costs in the graph satisfy the triangle inequality, we can solve the problem to optimality by decoupling the routing problem and the dwell time optimization problem. This assumption is reasonable since we desire the vehicles to revisit each POI frequently, and the longer the vehicle takes to revisit a particular POI, the lower the quality of information obtained from the vehicle.

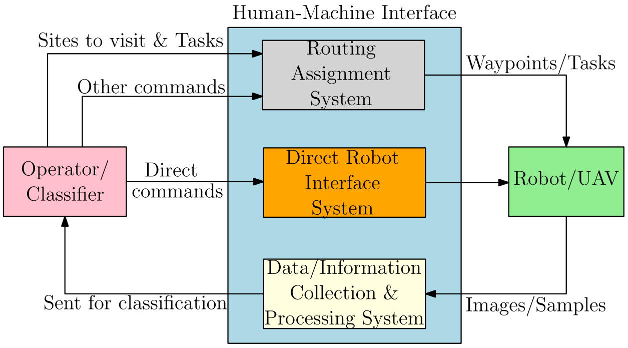


Fig. 1 Human-machine system with human as a classifier-in-the-loop

For the multi-vehicle problem, the partitioning of targets between the vehicles must also be considered, which increases the problem's difficulty. For this reason, we develop fast heuristics to obtain high-quality, feasible solutions for the considered problem under a simplifying assumption of triangle inequality and benchmark the obtained solutions with an algorithm for a similar problem.

The novel contributions of this paper are as follows:

- A novel model is proposed to account for the dependence of information gained on the UAV's dwell time and incorporate the cost of routing to enhance the operator's performance as a classifier-in-the-loop; maximization of the resulting discounted information gain is the objective of the proposed optimization scheme.
- The objective of routing is to maximize information gain, a nonlinear function of the continuous decision variables (dwell times) and discrete decision variables (routing choice). We show that the single-vehicle problem can be solved to optimality for the chosen objective function and obtain results for instances with at most 229 targets.
- We develop a fast heuristic for the multi-vehicle variant of the problem based on Variable Neighborhood Search [3], and perform substantial parametric studies on the choice of neighborhoods improving the incumbent solution over 43 instances.

The organization of the paper is as follows:

- In Section 2, we present a comprehensive literature review of routing problems, the use of information gain for classification, and the gap in the literature for considering the proposed problem.
- In Section 3, we present a mathematical representation for information gain and formulate the routing problem for information gain for a single vehicle and multiple vehicles. Furthermore, for the single vehicle case, we show that the problem can be solved to optimality.
- In Section 4, we introduce the heuristic considered to produce high-quality feasible solutions for the multi-vehicle problem and introduce various variations in the considered heuristic.
- We present the computational results for the single-vehicle and multi-vehicle cases in Section 5 and discuss improvements and trade-offs in the objective value and computation time for the different variations in the heuris-

tic considered. Furthermore, we compare the results of the heuristic with the current best-known heuristic for a similar vehicle routing problem in this section.

- Finally, we present the conclusions of this paper in Section 6.

Before proceeding to the literature review and the mathematical formulation, we will first describe an outline of the paper to provide the rationale behind the adopted methodology. The motivation of the considered problem is to route a set of m vehicles to cover n suspicious areas of activities provided by an operator and classify each Point of Interest as a target or not a target. To make this generic problem of practical significance concrete, an optimization problem is needed and formulated in this paper. In this regard, the first step in formulating an optimization problem is to formulate the objective function. We build up to the formulation of the objective function in Section 3, wherein we first derive the expression for information gain and discount it using the revisit time to incentivize the vehicle to visit other targets as well. The associated constraints pertain to routing the vehicle(s), wherein each target must be visited once, and a tour must be constructed for the vehicle(s). Furthermore, we show that the single-vehicle case can be solved to optimality, and we employ a branch-and-cut algorithm to solve the routing problem and utilize gradient descent to obtain the dwell time at each target.

Due to the difficulty of the multi-vehicle case stemming from a coupled routing and dwell-time optimization, we propose a heuristic to obtain a high-quality feasible solution. Since the studied problem is novel, we develop the heuristic step-by-step in Section 4, building on the basic neighborhood-based search technique proposed in [3] and a neighborhood-based technique developed for a min-max problem in [4].

Remark The code for the implementation of the algorithms to solve the single-vehicle and multi-vehicle problems and the instances considered are available at https://github.com/pranavraj575/UAV_routing_classification.

2 Literature Review

The topic of routing unmanned vehicles for autonomously collecting data has received significant attention in the literature. The routing objective depends on the mission, and the nature of the optimal solution depends further on the operational, motion, coordination, and communication constraints. Typically, the routing problems are fundamentally modeled as a variant of the popular traveling salesman problem (TSP) [5, 6], wherein a salesman/vehicle starts at a given location and must visit all locations/targets exactly once and return to the starting location with the cheapest tour cost. For example, routing of multiple vehicles starting at different initial locations has been addressed in [7] by developing an approximation ratio algorithm, and in [8, 9] using a graph transformation and a state-of-the-art heuristic, known as LKH [10], for the TSP. On the other hand, studies such as [11–13] address least-cost routing for problems wherein single or multiple vehicles with a bound on the heading rate, referred to as a Dubins vehicle [14], are considered. Studies such as [15, 16] address routing a vehicle to monitor a given set of targets persistently. More complicated routing problems involving multiple vehicles have also been studied recently, such as in [17], wherein the cost of communication between two vehicles is minimized to address vehicle routing under constant communication, and [18], wherein one vehicle assists the motion of another vehicle.

While the previous references addressed routing problems, none deal with routing vehicles to enhance an operator's performance as a classifier-in-the-loop. Typically, the performance of an operator-in-the-loop is specified by a confusion matrix, which specifies two conditional probability distributions – given that the event/object is of type $X \in \{T, F\}$, the conditional probability distribution specifies the discrete probability distribution of correctly classifying the event/object [19]. These probability distributions depend on controllable operational parameters such as altitude, pose of the vehicle relative to the object, time spent imaging the object/event, etc. The idea of vehicles enhancing the classification performance then rests on controlling or choosing these parameters to ensure that the conditional distributions are separated as far as possible, i.e., the mutual information gain is maximized. For persistent monitoring of the targets, minimizing the time, R, between successive revisits to the targets is also necessary. Discounting the mutual gain by a function of the revisit time is therefore reasonable, and the corresponding objective for optimization is to maximize the discounted mutual information gain over all the targets through optimal routing and determination of optimal dwell time.

Information gain [20] has been used in path planning in robotic applications that are distinct from what is considered in this paper. Lee et al. [21] developed an enhanced ant colony optimization for the capacitated vehicle routing problem by using information gain to ameliorate the search performance when a simulated annealing algorithm provided a good initial solution. Toit and Burdick [22] used the information gain theory in developing a partially closed-loop receding horizon control algorithm to solve the stochastic dynamic programming problem associated with dynamic uncertain environments robot motion planning. Kaufman et al. [23] presented a novel, accurate, and computationally efficient approach to predict map information gain for autonomous exploration where the robot motion is governed by a policy that maximizes the map information gain within its set of pose candidates. Zaenker et al. [24] proposed a novel view motion planner for pepper plant monitoring while minimizing occlusions (a significant challenge in monitoring large and complex structures), that builds a graph network of viable view poses and trajectories between nearby poses which is then searched by planner for graphs for view sequences with highest information gain. Paull et al. [25] used an information gain approach in the objective function of sidescan sonars (SSS) and for complete coverage and reactive path planning of an autonomous underwater vehicle. Mostofi [26] proposed a communication-aware motion-planning strategy for unmanned autonomous vehicles, where each node considers the information gained through both its sensing and communication when deciding on its next move. They showed how each node can predict the information gained through its communications by learning link quality measures online and combining it with the information gained through its local sensing in order to assess the overall information gain.

Information gain finds its place in machine learning literature, where it is being used for diverse feature ranking and feature selection techniques in order to discard irrelevant or redundant features from a given feature vector, thus reducing the dimensionality of the feature space. Novakovic [27] applied information gain for the classification of sonar targets, where the information gain evaluation helped in increasing computational efficiency while improving classification accuracy by doing feature selection.

Information-theoretic methods have been used in heuristics for path-planning methods in autonomous robotic exploration, where mutual information is calculated between the sensor's measurements and the explored map. Deng et al. [28] proposed a novel algorithm for optimizing exploration paths of a robot to cover unknown 2D areas by creating a gradient-based path optimization method that tries to improve the path's smoothness and information gain of uniformly sampled view-points along the path simultaneously. Julian et al. [29] proved that any controller tasked to maximize a mutual information reward function is eventually attracted to unexplored space, which is derived from the geometric dependencies of the occupancy grid mapping algorithm and the monotonic properties of mutual information. Bai et al. [30] proposed a novel approach to predict mutual information using Bayesian optimization to explore a priori unknown environments and produce a comprehensive occupancy map. They showed that the information-based method provides not only computational efficiency and rapid map entropy reduction but also robustness in comparison with competing approaches. Amigoni & Caglioti [31] presented a mapping system that builds geometric point-based maps of environments employing an information-based exploration strategy. The strategy determines the best observation positions by blending expected gathered information (that is measured according to the expected a posteriori uncertainty of the map) and the cost of reaching observation positions. Basilico & Amigoni [32] further extended this information-based exploration strategy for rescue and surveillance applications. In [33], the problem of routing mobile agents for data aggregation in sensor networks was considered. Here, the main issue was the trade-off between increasing information gain and power consumption among the source nodes that must be visited by the mobile agent, and was accounted for in the cost of the edges.

The problem of routing vehicles for aiding an operator-in-the-loop for classification was first proposed by Montez [34, 35]; however, the paper [34] does not exploit the exponential discounting nature of mutual information gain to decouple the mixed-integer nonlinear program into a discrete optimization problem and a continuous optimization problem. The form for the probability of correct classification of a POI as a target or not a target considered in [35] does not possess the desired structure for information gain that we seek in this article and is different from [35] in that respect. This paper exploits this structure, and an exact algorithm for single-vehicle routing is presented. In addition, the extension to the multiple vehicle case is presented with fast heuristics to generate high-quality feasible solutions, along with corroborating computational results.

3 Mathematical Formulation

3.1 Quantifying the Information Gained

Suppose a vehicle visits the i^{th} target. Denote the set of classification choices as $C = \{T, F\}$. Each POI has a correct classification $X \in C$. The operator assigns a classification of $Z \in C$ to the i^{th} POI after the visit. Let s_i represent the variables affecting the observation. Denote the conditional probabilities of correctly classifying i as T or F given the variables s_i as

$$P_t(s_i) = P(Z = T \mid X = T, s_i),$$
$$P_f(s_i) = P(Z = F \mid X = F, s_i),$$

respectively. Associated with the classification, we have a confusion matrix given in Table 1.

Table 1 Confusion matrix

	$Z = T$	$Z = F$
$X = T$	$P_t(s_i)$	$1 - P_t(s_i)$
$X = F$	$1 - P_f(s_i)$	$P_f(s_i)$

The two rows of the confusion matrix indicate the probability distribution of classification conditioned on the POI being of type T and F, respectively, and depend on the two parameters $P_t(s_i)$ and $P_f(s_i)$. Ideally, one would want to separate the two conditional probability distributions as much as possible with controllable variables that affect the observation, such as dwelling more time at a POI or observing the POI at a lower altitude or from a better perspective. The information gained by visiting each POI can be quantified using the Kullback-Leibler divergence (also referred to as mutual information or information gain) as the distance between the two conditional probability distributions. The mutual information for POI i between the two classification variables X and Z will be denoted as $I_i(X, Z)$. The mutual information is defined to be

$$I_i = \sum_{x,z \in C} P(X = x, Z = z) \log \frac{P(X = x, Z = z)}{P(X = x)P(Z = z)}. \quad (1)$$

Denote the *a priori* probability a POI is a true target, $P(X = T)$, as p. It will be assumed the *a priori* probability a POI is a target is 0.5. That is, there is effectively no known information about the targets before sending out the vehicle to investigate, so each POI is equally likely to be either a true target or a false target. Additionally, it will be assumed it is equally difficult to correctly classify the i^{th} POI as a true target or a false target. That is, $P_t(s) = P_f(s) = P_i(s)$ for any set of variables s_i that affect observation. Then, Eq. 1 reduces to

$$I_i = P_i(s_i) \log P_i(s_i) + (1 - P_i(s_i)) \log(1 - P_i(s_i)) + \log 2. \quad (2)$$

If $P_i(s_i) = P_i(d_i)$, a function of the dwell time, d_i, at the i^{th} POI, then one can express mutual information gain I_i at the i^{th} POI as an explicit function of the dwell time d_i. At this point, we observe the following desired properties of the mutual information gain function:

- The function $I_i(d_i)$ is monotonically increasing with d_i; essentially, the information gain increases with the time a vehicle spends at the i^{th} target. Hence, $\frac{\partial I_i}{\partial d_i} \geq 0$.
- Law of diminishing returns applies to the information gain, i.e., the marginal increase in information gain decreases with the dwell time. Hence, $\frac{\partial^2 I_i}{\partial d_i^2} \leq 0$.
- Information gained is always non-negative, i.e., $I_i(d_i) \geq 0$.

A consequence of these properties is that $I_i(d_i)$ is log-concave as $I_i(d_i)\frac{\partial^2 I_i}{\partial d_i^2} - (\frac{\partial I_i}{\partial d_i})^2 \leq 0$. It is also true that $J_0 = \sum_{i=1}^{n} I_i(d_i)$ is log-concave since

$$\left[\sum_{i=1}^{n} I_i(d_i)\right]\left[\sum_{j=1}^{n} \frac{\partial^2 I_j}{\partial d_j^2}\right] - \left[\sum_{i=1}^{n} \frac{\partial I_i}{\partial d_i}\right]^2 \leq 0.$$

A consequence of this observation is that one may employ gradient ascent to $\log(J_0(d_1, \ldots, d_n))$ to arrive at the optimum. In this paper, we model $P_i(s_i)$ as

$$P_i(s_i) = P_i(d_i) = 1 - \frac{1}{2}e^{-\sqrt{d_i/\tau_i}}, \quad (3)$$

where τ_i is a positive constant representing the sensitivity to the time spent at the i^{th} target. This form of $P_i(s_i)$ is consistent with the previous desired properties for information gain. Correspondingly, the information gain may be expressed as solely a function of d_i as

$$\begin{aligned} I_i(d_i) = &\left(1 - \frac{1}{2}e^{-\sqrt{d_i/\tau_i}}\right) \log\left(1 - \frac{1}{2}e^{-\sqrt{d_i/\tau_i}}\right) \\ &- \frac{1}{2}e^{-\sqrt{d_i/\tau_i}}\left(\log 2 + \sqrt{d_i/\tau_i}\right) + \log 2. \end{aligned} \quad (4)$$

A sample plot of information gain corresponding to $\tau_i = 0.5$ is given in Fig. 2.

Since we want to incentivize the vehicles to visit all targets, we discount the information gained by the revisit time, R_i, for the i^{th} target as follows:

$$\psi_i(d_i, R_i) = e^{-\alpha R_i} I_i(d_i), \quad (5)$$

where $\alpha > 0$ is a positive constant, R_i is the time duration between successive revisits to the i^{th} target.

Remark Discounting the information gained by revisit time is necessary since, without such a discounting, the optimal dwell times at each target will be infinite. This is because the information gain as a function of dwell time, which is

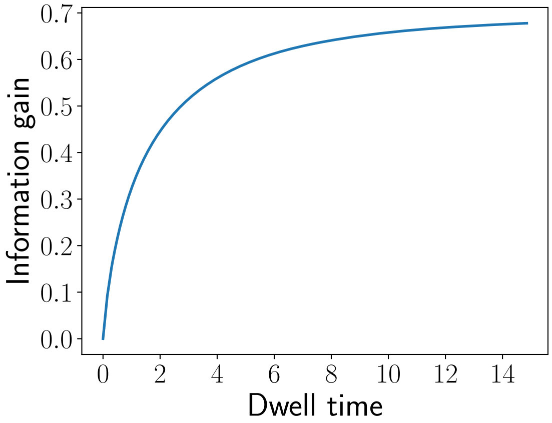


Fig. 2 Information gain vs dwell time at a POI ($\tau_i = 0.5$)

$I_i(d_i)$, is an increasing function, as can be observed from Fig. 2. In this regard, discounting incentivizes the vehicle to visit all targets required while ensuring that the total tour time obtained can be practically achieved. Furthermore, discounting the information gain also accounts for the cost of routing, which previously did not feature in the expression for information gain.

The objective of the optimization problem considered in this paper is to maximize

$$J_s(d_1, d_2, \dots, d_n) = \sum_{i=1}^{n} \psi_i(d_i, R_i), \tag{6}$$

through the choice of a vehicle route and a dwell time at each POI while ensuring that each target is visited. We note that since I_i is log-concave, J_s is also log-concave.

3.2 Single Vehicle Case

In the case of a single vehicle, R_i is the same for every target (say, it is R) if every other target is visited exactly once between successive revisits; moreover, $R = T_{tot} + \sum_{i=1}^{n} d_i$, where T_{tot} is the time taken to tour the n targets. Since triangle inequality is assumed to hold, this is true even if one may allow the same target to be visited multiple times between consecutive revisits to another target [15]. Note that $T_{tot} \geq TSP^*$, where TSP^* is the minimum time to visit the n targets before returning to the starting location. A consequence is the following:

$$\begin{aligned} e^{-\alpha R} &\leq e^{-\alpha TSP^*} e^{-\alpha \sum_{i=1}^{n} d_i}, \\ \implies J &\leq \sum_{i=1}^{n} e^{-\alpha TSP^*} e^{-\alpha \sum_{i=1}^{n} d_i} I_i(d_i), \\ &\leq e^{-\alpha TSP^*} \max_{d_1,\dots,d_n} e^{-\alpha \sum_{i=1}^{n} d_i} \sum_{i=1}^{n} I_i(d_i). \end{aligned}$$

If J^* is the optimum, clearly, it is achieved by minimizing T_{tot} and maximizing the log-concave function on the right-hand side of the above inequality. In other words, the problem of optimal routing and determining optimal dwell time at each target is now decoupled and can be solved optimally by solving the former problem with an integer program formulation and the latter problem using gradient descent.

3.3 Multiple Vehicle Case

An additional complication arises in the multiple vehicle case - partitioning and assigning the targets to be visited by each vehicle. If there are $m > 1$ vehicles, let the targets be partitioned into m *disjoint* sets, namely $\mathcal{P}_1, \dots, \mathcal{P}_m$, so that the j^{th} vehicle is tasked with visiting the POIs in $\mathcal{P}_j$. Let R_j be the revisit time associated with targets assigned to the j^{th} vehicle, and the associated tour cost for persistent monitoring per cycle be $TSP^*(\mathcal{P}_i)$. Associated with the j^{th} vehicle, the discounted information gained is given by

$$\begin{aligned} J_j(\mathcal{P}_j) &= \max_{d_i, i \in \mathcal{P}_j} \left(e^{-\alpha R_j} \sum_{i \in \mathcal{P}_j} I_i(d_i) \right) \\ &= e^{-\alpha TSP^*(\mathcal{P}_j)} \max_{i \in \mathcal{P}_j} \left(e^{-\alpha \left(\sum_{i \in \mathcal{P}_j} d_i \right)} \sum_{i \in \mathcal{P}_j} I_i(d_i) \right). \end{aligned} \tag{7}$$

Corresponding to the partitions $\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_m$, the discounted information gain is given by

$$J(\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_m) = \sum_{j=1}^{m} J_j(\mathcal{P}_j), \tag{8}$$

where $J_j(\mathcal{P}_j)$ is given in Eq. 7. Correspondingly, the objective is to maximize the discounted information gain over all possible partitions, sequences of visiting POIs by every vehicle, and the dwell time at each target:

$$J = \max_{\mathcal{P}_j,\, 1 \leq j \leq m} J(\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_m). \tag{9}$$

Since maximizing over partitions is a difficult combinatorial problem, we provide heuristics for the outer layer of optimization in the above optimization problem and use the single vehicle algorithm for the inner layer of optimization.

4 Heuristic for Multi-Vehicle Case

From the previous section, it can be observed that the single-vehicle case can be solved to optimality. For the multi-vehicle case, the problem is a coupled routing problem and continuous optimization problem stemming from dwell-time optimization. Hence, in this paper, a neighborhood-search based heuristic [3] is proposed to obtain high-quality solutions for the multi-vehicle case. To this end, a heuristic inspired by the MD algorithm [4], which is a heuristic that yields high-quality solutions for a min-max multi-vehicle multi-depot problem, is discussed. The intuition behind using a similar heuristic structure for a min-max problem for the proposed problem is as follows:

- From the objective function, a vehicle having a high tour cost will have a low objective value due to the high penalty incurred due to the exponential term (the discounting term).

- Further, a vehicle visiting very few targets will have a low objective value due to very minimal information gain.

Hence, similar to the min-max problem, it is desired that vehicle tours be generated so that the vehicles are load-balanced.

The heuristic is structured through three steps: (i) generation of an initial feasible solution, (ii) local search, and (iii) perturbation of solution. The following subsections will expand the three steps in the heuristic. For this purpose, the notation used for the graph will be discussed first. Let T denote the set of targets to be covered by m vehicles in a graph G. Let the j^{th} vehicle start at depot D_j for $j \in \{1, 2, \cdots, m\}$. Hence, the vertices of graph G are $T \bigcup \{D_1, D_2, \cdots, D_m\}$. We note here that the depots need not be necessarily distinct. The set of edges E in graph G are assumed to be symmetric, complete, and satisfying triangle inequality (as previously stated). Further, let c_{kl} (and $c(k, l)$) denote the Euclidean distance between vertices k and l in the graph. Without loss of generality, the vehicles are considered to be traveling at a unit speed. Hence, c_{kl} denotes the cost of edge $(k, l) \in E$.

4.1 Initial Feasible Solution

The initial feasible solution is generated using a load balancing technique studied in [36] and subsequently utilized in the MD algorithm. In this technique, an assignment problem is formulated. Consider the depots of the vehicles, which are indexed by j, and the targets, which are indexed by i. Consider a binary variable x_{ij} denoting whether target i is allocated to depot j or not. If target i is allocated to depot j, then $x_{ij} = 1$, and if not, $x_{ij} = 0$. Let c_{ij} denote the Euclidean distance between target i and depot j. Consider the following integer program formulation:

$$\min \sum_{i=1}^{|T|} \sum_{j=1}^{m} c_{ij} x_{ij} \tag{10}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} x_{ij} = 1, \quad \forall i \in T, \tag{11}$$

$$\sum_{i=1}^{|T|} x_{ij} = \begin{cases} \left\lceil \frac{|T|}{m} \right\rceil, & j = 1, 2, \cdots, (|T| \bmod m), \\ \left\lfloor \frac{|T|}{m} \right\rfloor, & j = (|T| \bmod m) + 1, \cdots, m, \end{cases} \tag{12}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in T \quad \forall j \in \{1, 2, \cdots, m\}. \tag{13}$$

In the above formulation, constraint (11) ensures that each target is allocated to exactly one vehicle. Constraint (12) ensures that each vehicle is allocated approximately the same number of targets. Since the number of targets can be expressed as $|T| = pm + q$, where p and q are integers, the first q number of vehicles are allocated $p + 1 = \left\lceil \frac{|T|}{m} \right\rceil$ number of targets, and the other $m - q$ number of vehicles are allocated $p = \left\lfloor \frac{|T|}{m} \right\rfloor$ number of targets. In the above formulation, the objective function in Eq. 10 allocates targets to depots that are in its vicinity. It should be noted that the above formulation can be solved as a linear program to obtain the optimal solution for the integer program.

It should be noted that if multiple vehicles start from the same depot location, then the depot location is perturbed about its initial location for each vehicle, similar to the MD algorithm [4]. If the perturbation is not performed, varying target allocations between the vehicles that start at the same location results in no change in the objective function. Similar to [4], the initial locations were placed symmetrically on a circle of radius 0.1 centered at the initial depot location (before perturbation). The angle of one of the perturbations was chosen randomly, and the perturbation angles for the other vehicles starting at the same depot were obtained such that perturbed depots were placed symmetrically on the circle.

4.2 Local Search

A local search is performed to improve the solution obtained by seeking better solutions in a neighborhood around the incumbent solution [3]. For this purpose, two neighborhoods are considered: Neighborhood 1 and Neighborhood 2. In both neighborhoods, a vehicle is first chosen.

- In the first neighborhood (N1), a target is attempted to be removed from this vehicle and inserted into another vehicle. A depiction of N1 is shown in Fig. 3.
- In the second neighborhood (N2), a target from the chosen vehicle is attempted to be swapped with a target allocated to another vehicle.

For the implementation of these neighborhoods and to ensure that the computations are fast, three questions need to be answered:

- How do we pick the vehicle to remove a target from (referred to as the maximal vehicle)?
- How do we sort the list of targets in the vehicle from which the target is attempted to be removed?
- How do we pick the vehicle in which a removed target needs to be inserted in (in N1) or swapped with (in N2)? (For reference, the vehicle picked for insertion in Fig. 3 is the “blue” vehicle.)

To answer these questions, two proxy costs will be defined: a cost associated with removing a target from a vehicle and a cost associated with inserting a target into a vehicle.

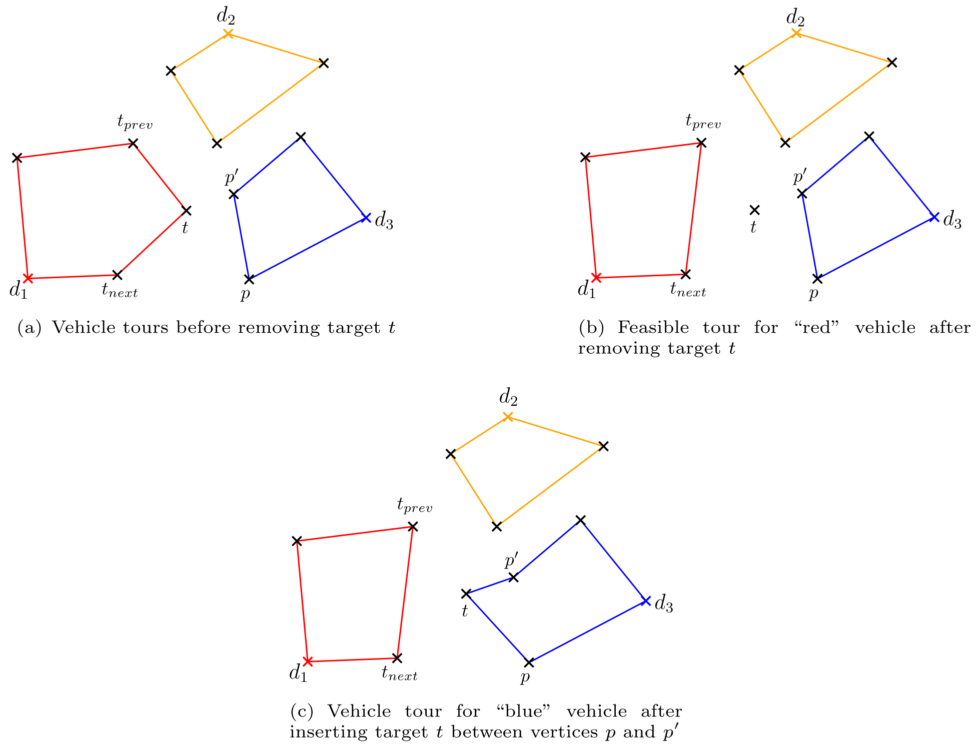


Fig. 3 Vehicle tours after removing target t from red vehicle and inserting into "blue" vehicle

(a) Vehicle tours before removing target t

(b) Feasible tour for "red" vehicle after removing target t

(c) Vehicle tour for "blue" vehicle after inserting target t between vertices p and p'

4.2.1 Proxy Cost for Target Removal

Consider removing a target t from the vehicle shown in red, the maximal vehicle, in Fig. 3. It is desired to estimate the increase in the objective value of the red vehicle (denoted with index j) after removing target t from it. Let t_{prev} and t_{next} denote the targets covered before and after target t in the maximal vehicle's tour, and the maximal vehicle's tour be given by $(u_0 = D_j, u_1^j, u_2^j, \cdots, t_{prev}, t, t_{next}, \cdots, u_{n_j-1}^j, u_{n_j}^j = D_j)$. It should be noted here that D_j denotes the depot of the maximal vehicle. Hence, the estimated increase in cost associated with t is defined to be

$$\begin{aligned}
&\text{increase in objective}_t \\
&= \text{estimated new objective}_j - \text{old objective}_j \\
&= J_j^{est}\left(d_{u_1^j}, d_{u_2^j}, \cdots, d_{t_{prev}}, d_{t_{next}}, \cdots, d_{u_{n_j-1}^j}\right) \qquad (14)\\
&\quad - J_j\left(d_{u_1^j}, d_{u_2^j}, \cdots, d_{t_{prev}}, d_t, d_{t_{next}}, \cdots, d_{u_{n_j-1}^j}\right),
\end{aligned}$$

where d_v denotes the dwell time for a vertex v. Here, $J_j^{est}\left(d_{u_1^j}, d_{u_2^j}, \cdots, d_{t_{prev}}, d_{t_{next}}, \cdots, d_{u_{n_j-1}^j}\right)$ denotes the estimated new objective for the j^{th} vehicle after removing target t. For computing the estimated objective value, estimates for the dwell times of the other targets after removing target t, and the cost associated with the tour (TSP) after removing t, need to be obtained. To this end,

- The dwell times of targets $u_1^j, u_2^j, \cdots, t_{prev}, t_{next}, \cdots, u_{n_j-1}^j$ are maintained to be the same as before removing target t.
- Further, for the estimated tour cost, a "savings" metric similar to the MD algorithm [4] was considered. The savings corresponding to target t is given by

  $$\text{savings}_t = c(t_{prev}, t) + c(t, t_{next}) - c(t_{prev}, t_{next}).$$

  Hence, the new tour cost is given by the previous tour cost minus the savings corresponding to target t.

4.2.2 Proxy Cost for Target Insertion

Similar to the "increase in objective" metric associated with removing a target, it is desired to define an "increase in objective insert" metric associated with inserting target t in the "blue" vehicle shown in Fig. 3. This metric estimates an increase in the objective value obtained by inserting target t in the "blue" vehicle (denoted with i henceforth). For vehicle i,

- The dwell times of the other targets covered by the vehicle, which are $u_1^i, \cdots, p, p', \cdots, u_{n_i-1}^i$ are maintained to be the same as before inserting target t. Further, the dwell time of target t before removing from vehicle j and after inserting in vehicle i is kept to be the same.
- Suppose t is desired to be inserted between targets p and p' in the i^{th} vehicle's tour. The insertion cost associated

with inserting t in vehicle i between targets p and p' is defined to be

$$\text{cost increase}_{t,i,(p,p')} = c(p,t) + c(t,p') - c(p,p').$$

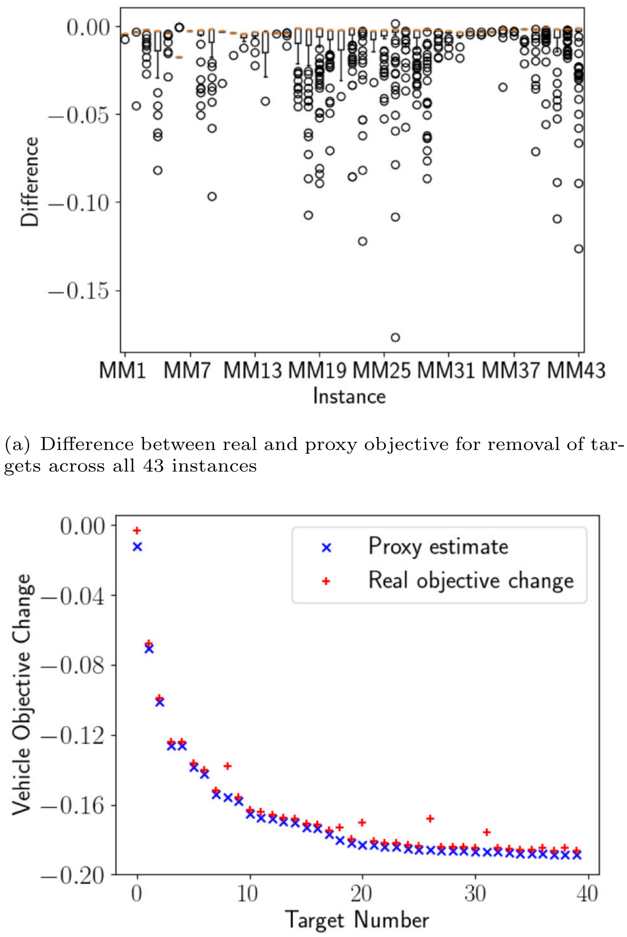
This metric estimates the increase in the tour cost (TSP) with the insertion of t between targets p and p'. Since target t can be inserted between any pair of vertices in vehicle i's tour, the "cost increase" for vehicle i for inserting target t is defined to be the minimum among all such costs. This choice of insertion will also yield the highest estimated increase in the objective value (as can be observed from Eq. 7) since the discounting due to the exponential will be the least. The final insertion location of t corresponds to "cost increase".

The "increase in objective insert" corresponding to inserting target t for vehicle i can therefore be expressed as

$$\begin{aligned}
&\text{increase in objective insert}_{t,i} \\
&= \text{estimated new objective}_i - \text{old objective}_i \\
&= J_i^{est}\left(d_{u_1^i}, d_{u_2^i}, \cdots, d_t^{est}, \cdots, d_{u^i_{n_i-1}}\right) \qquad (15) \\
&\quad - J_i\left(d^i_{u_1}, d_{u_2^i}, \cdots, d_{u^i_{n_i-1}}\right).
\end{aligned}$$

4.2.3 Analysis of Proxy Costs

To justify using our proxy objective estimates in local search, we compare the estimated change in vehicle objective with the real change. The initial feasible solution based on the assignment problem is first generated for each instance. We consider the removal of all targets from the vehicle corresponding to the highest tour cost since such a vehicle can correspond to low objective value due to high discounting. The results in Fig. 4(a) (shown as a box plot) show the difference between the true objective value, obtained using LKH [10], a state-of-the-art heuristic for the TSP, and the proxy estimate of the chosen vehicle over 43 instances taken from [4]. It should be noted that the instances considered are diverse with respect to the number of targets, the number of vehicles, and the probability distribution from which the target locations were obtained. From this figure, we can observe that the proxy costs provide a relatively accurate estimate of the objective value with target removal. We note that though all values in the box plot should be non-positive, we see a few targets where the proxy estimate is larger than the real objective, which arises due to rounding of edge costs for running LKH. We also show the true objective change and estimated objective change for one of the instances for all targets of the vehicle with the highest tour cost in Fig. 4(b). We can observe that the true objective change and the proxy estimate match closely, justifying the proxy cost choice.



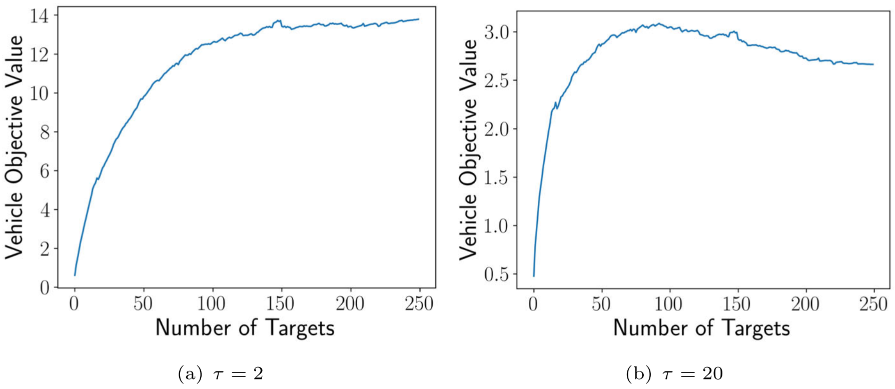
(a) Difference between real and proxy objective for removal of targets across all 43 instances

(b) Comparison of real objective change and proxy objective change on MM30

Fig. 4 Proxy objective estimates compared to the real tour objective obtained through LKH and gradient descent

4.2.4 Variations of Vehicle Choice Considered for Removal of Target

Having defined the estimated increase in the objective value associated with removing and inserting a target, it is desired to identify the vehicle (termed the maximal vehicle) from which a target is attempted to be removed (the "red" vehicle in Fig. 3). While the choice of the maximal vehicle is immediate in a min-max problem, which is the vehicle that has the highest tour cost, such a choice of maximal vehicle is not immediate for our considered problem. To this end, four variations in the decision of the vehicle to be picked were considered. While one of the variations described below will be a natural choice for this problem, the justification for three



(a) $\tau = 2$

(b) $\tau = 20$

Fig. 5 Sample vehicle objective value variation with number of targets for instance MM22 (from [4]) for $\alpha = 8.77 \cdot 10^{-4}$, $\tau = 2$ and $\alpha = 8.77 \cdot 10^{-4}$, $\tau = 20$

of the variations is drawn using Fig. 5. In this figure, the variation in the objective value of the first vehicle, obtained from Eq. 7, with changing number of targets is shown for instance MM22 (taken from [4]). The value for α was computed as $\frac{1}{TSP}$, where TSP denotes the tour cost to cover all targets by a single vehicle, which was obtained using LKH.[1] Further, two variations in τ were considered for this plot.

The variations considered in the choice of maximal vehicle and the reason behind considering each variation are as follows:

- Vehicle with minimum objective value: The intuition behind such a vehicle choice is that a vehicle with a low objective value will be indicative of a vehicle with a high tour cost. This, in turn, would lead to a high discounting value due to the exponential in the objective function (refer to Eq. 6).
- Vehicle with maximum objective value: The intuition behind removing a target from such a vehicle is that the considered vehicle's objective value would not significantly be affected by removing a target, as shown in Fig. 5. However, inserting the removed target in another vehicle can make a net improvement in the objective value.
- Vehicle with the maximum number of targets: If a vehicle visits many targets, then the removal of a target can lead to an increase in the objective value due to reduced discounting due to the exponential function (refer to Eq. 6 and Fig. 5).
- Vehicle with maximum tour cost: The reason for this choice is the same as the choice of vehicle with the maximum number of targets. However, this variation is considered since a vehicle with the maximum tour cost does not necessarily need to be the same as the vehicle with the maximum number of targets.

It is desired to determine which of the variations will yield a good improvement with respect to the initial solution. To this end, it is desired to study the influence of vehicle choice on the objective value obtained after a local search for different instances. For this purpose, we will first describe neighborhood 1, which will later be used for this study (and for the heuristic).

4.2.5 Neighborhood 1: 1 pt. Move

For a chosen vehicle ("maximal" vehicle) from which a target is desired to be removed, an order of considering targets in the maximal vehicle is first desired to be constructed. To this end, the "increase in objective" metric associated with removing a target, given in Eq. 14, is used. Since removing a target that can provide a high increase in the objective value is desired, the targets in the maximal vehicle are sorted in the decreasing order of this metric.

Suppose target t is considered to be removed from the maximal vehicle. The vehicle in which the target t will be inserted needs to be obtained. To this end, the "increase in objective insert" metric previously defined is used. Among all vehicles other than the maximal vehicle, the vehicle with the highest estimated increase in the objective value is chosen for inserting target t (since a maximization problem is considered). Hence, if the estimated objective value corresponding to the new solution obtained after removing t from the maximal vehicle and inserting it in another vehicle is higher than the previous objective value, then the obtained solution is chosen as the new incumbent solution. Further, LKH is used to construct the tours of the maximal vehicle and the vehicle considered for insertion, and gradient descent is used to obtain the dwell time for each target covered by the two vehicles. If a better solution was not obtained by removing target t, the next target in the sorted list by the "increase in objective" metric defined is considered from the maximal vehicle. The same steps are performed until a better solution is obtained or all targets from the maximal vehicle are considered.

[1] The reason for considering such an α value based on the TSP cost was to ensure that the discounting factor corresponding to the exponential in Eq. 7 is not very large. In such cases, it was observed that one of the vehicles covered all targets, whereas other vehicles covered no targets or one target.

Remark We note here that if a better solution is obtained based on the proxy costs compared to the incumbent solution, then rerunning LKH and gradient descent for the new allocation of targets will yield a solution with an objective value that is greater than or equal to the estimated objective value based on proxy costs. This is because we always obtain a feasible solution for the vehicles using our proxy cost definition and target reallocation. This remark is valid for Neighborhood 2 as well, which we define later.

4.2.6 Impact of Maximal Vehicle Considered for the Local Search

For the neighborhood previously considered, a maximal vehicle needs to be identified, which is the vehicle from which a target is attempted to be removed. For this purpose, four variations were considered, as previously mentioned. The percentage improvement from the local search using neighborhood 1 with respect to the initial solution for all four methods are reported for 43 instances, which were taken from [4]. The obtained percentage improvement for the four variations are shown in Fig. 6, and the minimum, median, mean, and maximum percentage improvements are summarized in Table 2. It can be observed that the variations corresponding to choosing the vehicle with the most number of targets and the vehicle with the highest tour cost yielded the best improvement with respect to the initial solution.

Since both variations yield a good improvement with respect to the initial solution and different improvement values, the previous neighborhood considered, i.e., N1, was modified to consider two types of "maximal" vehicles. First, the maximal vehicle is considered to be the vehicle with the longest tour. If a better solution was obtained, then the solution obtained would be considered the new incumbent solution. If no improvement can be obtained, then the same neighborhood search is performed, but with the maximal vehicle considered to be the vehicle with the most targets. The percentage improvement with this modified neighborhood is also reported in Fig. 6 and Table 2, and can be observed to provide a better solution than the solutions previously obtained. Hence, the modified neighborhood considering both types of maximal vehicles would be considered for the heuristic.

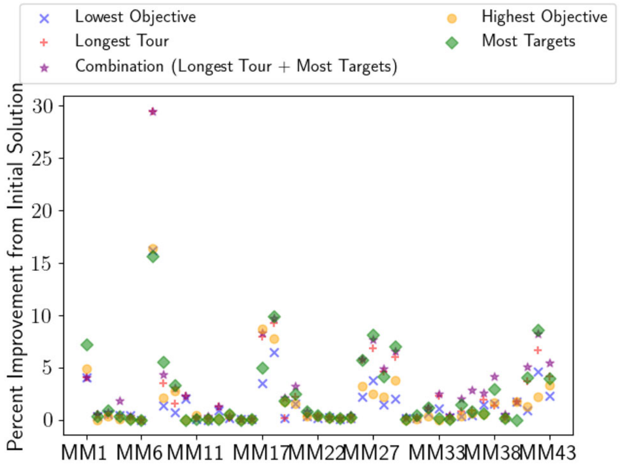


Fig. 6 Percentage improvement using 1 pt. move by heuristic for "vehicle to remove target from" based on different vehicle choices

Table 2 Percentage improvement of local search by heuristic for 'vehicle to remove target from'

Method	Min.	Median	Mean	Max.
Lowest objective	0	0.5	1.5	16.2
Highest objective	0	0.33	1.68	16.37
Most targets	0	0.61	2.43	15.64
Longest tour	0	0.88	2.64	29.47
Combination (Longest tour + Most targets)	0	1.86	3.16	29.47

4.2.7 Neighborhood 2: 1 pt. swap

Using the identified combination of maximal vehicles in neighborhood 1, the same combination of maximal vehicles is used for neighborhood 2. In this neighborhood, it is desired to select a target from a maximal vehicle and swap it with a target from another vehicle to obtain a better solution. Using the observation of the combination of maximal vehicles previously considered, first

- The vehicle corresponding to the longest tour is utilized as the maximal vehicle, and neighborhood 2 is used.
- If a better solution was not obtained, the vehicle corresponding to the maximum number of targets is utilized as the maximal vehicle, and neighborhood 2 is used.

Having chosen the maximal vehicle (either the vehicle with the longest tour or the maximum number of targets), the targets are ordered in the maximal vehicle in the descending order of the increase in objective metric given in Eq. 14. Suppose target t is removed from the maximal vehicle. Similar to neighborhood 1, the target is inserted in the vehicle with the highest increase in the objective value corresponding to target t, defined in Eq. 15. Let the vehicle in which t is inserted be denoted by i. It is now desired that a target from vehicle i be removed and inserted into the maximal vehicle.

Similar to target removal from the maximal vehicle, the targets in vehicle i are ordered in the decreasing order of the increase in objective value given in Eq. 14. However, it should be noted that contrary to target removal from the maximal vehicle, the objective value of vehicle i before removing a target is an estimate. That is, in Eq. 14, "old objective$_i$" is the estimated objective value of vehicle i obtained after inserting t. Having ordered the targets in vehicle i in the decreasing order of the estimated increase in objective, each target is attempted to be inserted in the maximal vehicle. Suppose t_i is attempted to be inserted in the maximal vehicle's tour. For this purpose, the increase in the objective value corresponding to the insertion of t_i is obtained for the maximal vehicle, using Eq. 15. If a better solution is obtained, LKH is used to optimize the vehicle tours, and the dwell times are obtained using gradient descent for the two vehicles. The same steps are performed until a better solution is obtained or all targets from the maximal vehicle are considered.

4.3 Perturbation of Solution

The solution obtained from local search is a local minimum and cannot be improved using the defined neighborhoods. Hence, it is desired to break from the local minimum. To this end, the solution is perturbed, similar to the MD algorithm [4], by perturbing the depot locations. For this purpose, for the j^{th} vehicle, the average distance r_j of the depot D_j from the two targets connected to it in the vehicle's tour in the current solution is obtained for $j = 1, 2, \cdots, m$. Then, in the first round of perturbation, each depot location is perturbed from its initial location by a random angle to a location that is at a distance r_j from its initial location, as shown in Fig. 7. Using the same allocation of targets and the corresponding dwell times from the solution obtained from the local search for each vehicle, a feasible solution is generated for each vehicle. A local search is then performed on the graph with the perturbed depot locations to obtain a new allocation of targets for each vehicle and corresponding dwell times. Using this new solution in the original graph, wherein the depots are restored to their initial locations, a local search is then performed to obtain a new potential solution. If the obtained solution is better than the incumbent solution, then the incumbent solution is updated, and the perturbation step is restarted.

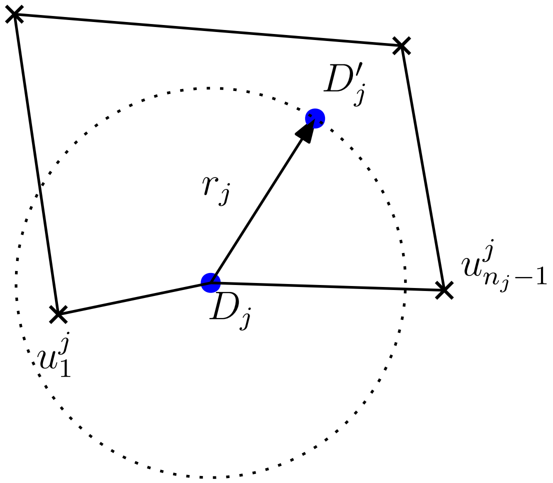


Fig. 7 Depiction of perturbation for the j^{th} vehicle

Similar to [4], the perturbation step is performed for five consecutive iterations until no improvement is obtained. The perturbation angle for each depot is set to be 144° from the previous iteration's perturbation angle. Hence, the sixth perturbation will be the same as the first perturbation.

A summary of the steps in the proposed heuristic is shown in Fig. 8.

5 Simulation Results

The results we present in this section were generated using Python 3.8 and Julia 1.9.2 on a laptop with AMD Ryzen 5 4600H running at 3 GHz with 8 GB RAM. We first present results corresponding to the single vehicle case, which can be solved to optimality. The multi-vehicle case results are then presented, wherein the performance of different variations of the presented heuristic are analyzed on a diverse set of instances.

5.1 Single Vehicle Results

The single-vehicle case was solved to optimality in two steps: solving the integer program corresponding to the single-vehicle TSP using branch and cut, and obtaining the dwell times using gradient descent. To this end, branch and cut was implemented in Julia using the JuMP package, and the integer program was solved using Gurobi [37]. The integer program was warm-started using the solution from LKH [10]. Further, for the branch and cut implementation, subtour elimination constraints were used as cutting planes and lazy cuts using lazy callback functions.

Six instances considered were taken from standard US datasets from TSPLIB [38]. The number of vertices in the instances varied from 100 to 229. For our simulations, we selected the first vertex to be the depot of the vehicle and the other vertices to be points of interest/targets. For each of the instances, three values of α were considered. The α values chosen were equal to $\frac{0.5}{TSP}, \frac{1}{TSP}, \frac{2}{TSP}$, where TSP here denotes the tour cost associated with covering all the targets in the graph, which was obtained using LKH. The reason for choosing such α's was to ensure that the discounting associated with the exponential terms in the objective function is not very high.

Remark When the α was set to larger values, i.e., when the discounting due to the exponential is high, it was observed that the optimal solution for three vehicle problems with ten targets was such that one of the vehicles covers a majority of the targets, and the other vehicles cover at most one target.

Fig. 8 Flowchart illustrating the heuristic considered for the multi-vehicle case

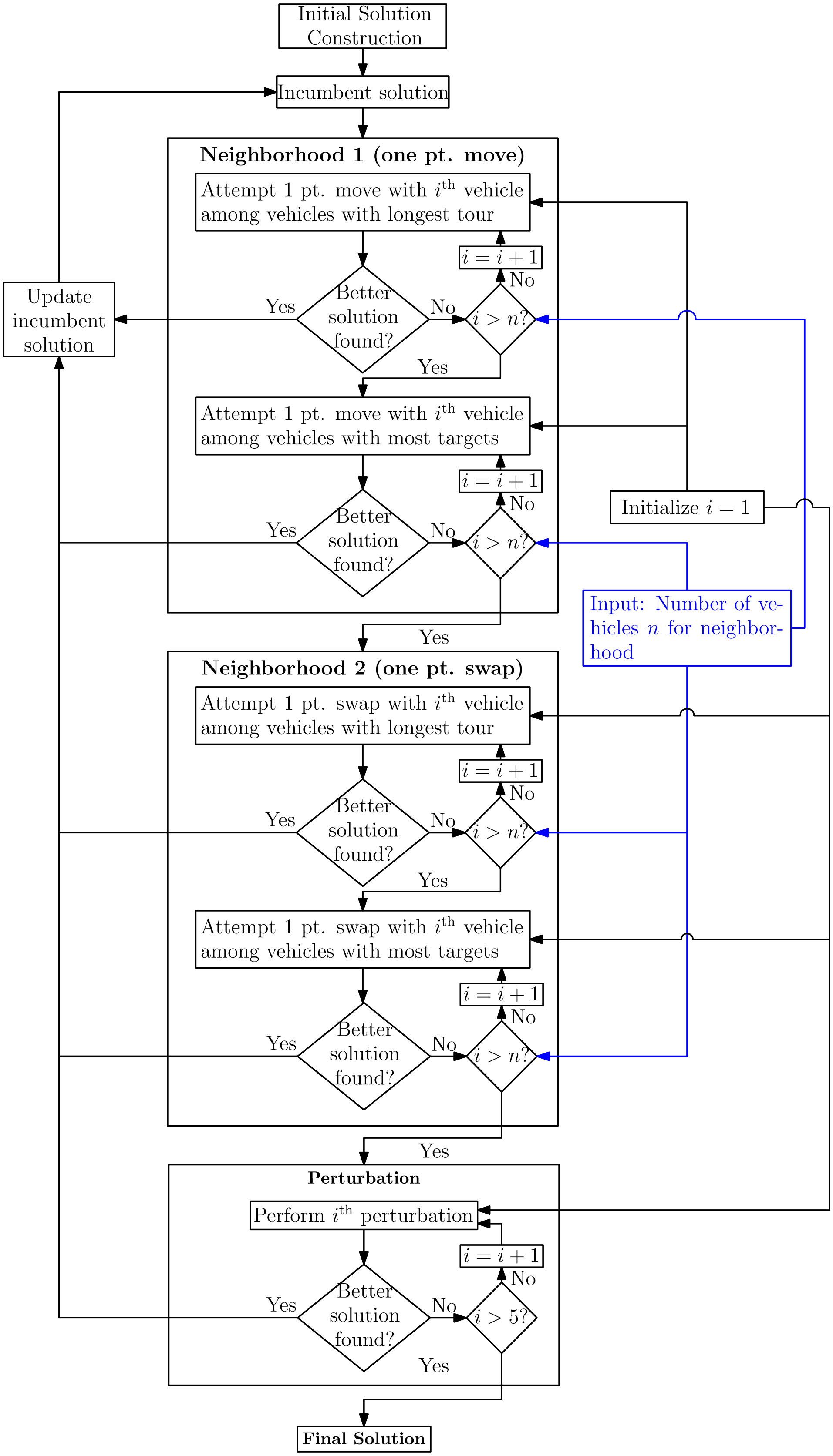

Initial Solution Construction
Incumbent solution
Neighborhood 1 (one pt. move)
Attempt 1 pt. move with i^{th} vehicle among vehicles with longest tour
Better solution found?
Yes
No
$i > n$?
$i = i + 1$
Update incumbent solution
Attempt 1 pt. move with i^{th} vehicle among vehicles with most targets
Initialize $i = 1$
Input: Number of vehicles n for neighborhood
Neighborhood 2 (one pt. swap)
Attempt 1 pt. swap with i^{th} vehicle among vehicles with longest tour
Attempt 1 pt. swap with i^{th} vehicle among vehicles with most targets
Perturbation
Perform i^{th} perturbation
$i > 5$?
Final Solution

Table 3 Results for single vehicle case

Instance	TSP cost*	α (s^{-1})	τ (s)	Dwell time (s)				Branch and cut time (s)	Gradient descent time (s)
				Min	Median	Mean	Max		
rd100	7910.4	6.37e-05	0.5	11.37	11.37	11.37	11.38	1.09	1.50e-02
			1.0	16.97	17.03	17.01	17.03	1.19	9.00e-03
			2.0	24.4	24.49	24.55	24.64	1.1	1.10e-02
		1.27e-04	0.5	8.76	8.77	8.78	8.78	1.08	5.00e-03
			1.0	12.65	12.75	12.72	12.76	1.11	6.00e-03
			2.0	17.56	17.82	17.71	17.85	1.13	8.00e-03
		2.55e-04	0.5	6.45	6.47	6.48	6.5	1.13	3.00e-03
			1.0	8.96	9.09	9.04	9.11	1.12	3.00e-03
			2.0	12.01	12.06	12.1	12.22	1.06	4.00e-03
bier127	118293.52	4.27e-06	0.5	19.16	19.16	19.16	19.17	1.04	2.10e-02
			1.0	30.45	30.45	30.45	30.46	1.05	3.10e-02
			2.0	47.1	47.16	47.17	47.19	1.04	4.50e-02
		8.54e-06	0.5	17.52	17.52	17.52	17.52	1.06	1.80e-02
			1.0	27.57	27.58	27.58	27.58	1.04	2.30e-02
			2.0	42.14	42.21	42.22	42.25	1.05	4.00e-02
		1.71e-05	0.5	15.3	15.3	15.31	15.31	1.05	1.40e-02
			1.0	23.7	23.72	23.72	23.72	1.06	1.90e-02
			2.0	35.57	35.64	35.67	35.72	1.05	2.50e-02
pr152	73683.64	6.79e-06	0.5	17.37	17.37	17.37	17.38	49.27	2.20e-02
			1.0	27.31	27.31	27.32	27.32	51.26	3.90e-02
			2.0	41.68	41.8	41.77	41.81	49.28	5.00e-02
		1.36e-05	0.5	15.3	15.31	15.3	15.31	49.38	1.80e-02
			1.0	23.69	23.72	23.71	23.72	49.57	4.40e-02
			2.0	35.54	35.64	35.65	35.69	52.36	4.10e-02
		2.72e-05	0.5	12.83	12.84	12.84	12.84	53.94	1.50e-02
			1.0	19.45	19.49	19.48	19.5	53.34	3.10e-02
			2.0	28.45	28.5	28.59	28.66	50.7	2.60e-02
d198	15808.65	3.17e-05	0.5	11.15	11.16	11.16	11.16	11.07	1.70e-02
			1.0	16.61	16.68	16.66	16.68	11.24	3.50e-02
			2.0	23.83	23.84	23.99	24.08	11.56	2.70e-02
		6.34e-05	0.5	8.67	8.7	8.69	8.7	11.07	1.20e-02
			1.0	12.5	12.56	12.58	12.62	11.31	1.60e-02
			2.0	17.34	17.36	17.5	17.62	11.02	1.90e-02
		1.27e-04	0.5	6.42	6.45	6.45	6.47	11.48	9.00e-03
			1.0	8.93	8.99	9.01	9.07	11.53	1.10e-02
			2.0	11.96	12.16	12.03	12.17	11.38	1.40e-02
pr226	80370.26	6.23e-06	0.5	15.95	15.95	15.95	15.95	34.36	4.40e-02
			1.0	24.82	24.82	24.84	24.84	32.13	6.10e-02
			2.0	37.45	37.49	37.56	37.6	32.17	7.90e-02
		1.25e-05	0.5	13.89	13.89	13.89	13.9	32.8	4.00e-02
			1.0	21.25	21.3	21.29	21.3	30.36	3.70e-02
			2.0	31.46	31.47	31.59	31.66	35.14	5.30e-02
		2.49e-05	0.5	11.49	11.5	11.49	11.5	34.81	2.20e-02
			1.0	17.16	17.23	17.21	17.23	33.04	4.50e-02
			2.0	24.73	24.78	24.88	24.98	33.16	5.80e-02

Table 3 continued

Instance	TSP cost*	α (s^{-1})	τ (s)	Dwell time (s)				Branch and cut	Gradient
				Min	Median	Mean	Max	time (s)	descent time (s)
gr229	1635.31†	3.06e-04	0.5	3.77	3.79	3.8	3.84	7.2	7.00e-03
			1.0	4.96	4.98	4.98	5.04	7.13	8.00e-03
			2.0	6.24	6.36	6.37	6.75	7.13	9.00e-03
		6.13e-04	0.5	2.5	2.5	2.51	2.54	7.25	4.00e-03
			1.0	3.12	3.15	3.16	3.21	7.34	3.50e-02
			2.0	3.7	3.82	3.8	3.83	7.81	1.40e-02
		1.23e-03	0.5	1.58	1.59	1.59	1.6	7.02	9.00e-03
			1.0	1.88	1.89	1.91	1.92	6.92	7.00e-03
			2.0	2.21	2.25	2.23	2.26	7.19	2.40e-02

*: If we calculate these by rounding each edge weight, we get results consistent with the known optimal solution for these instances

†: The known optimal solution for this instance corresponds to an edge cost computation different from Euclidean distance

For the instances, three values of τ were considered: 0.5, 1, and 2. The results obtained for the single vehicle case are summarized in Table 3. We remark here that the instance's name indicates the number of targets in the graph. From this table, we can observe that

1. The optimal solution for all instances could be obtained within about 30 seconds.
2. The dwell time for each target is approximately the same due to the same τ value for all targets.
3. Increasing τ increases the dwell time for each target, whereas increasing α decreases the dwell time. This is because a higher τ value necessitates the vehicle to stay at a target for a longer time to obtain the same information gain, as can be observed from the equation corresponding to the probability of classification given in Eq. 3. However, increasing α decreases the dwell time since a higher α penalizes a larger revisit time for the vehicle, as can be observed from the discounted information gain, which is given in Eq. 5.

Remark Through the presented results, it can be observed that single-vehicle case problems can be solved to optimality in an efficient manner.

5.2 Multi-vehicle Case

The instances considered in this study were taken from the MD datasets [4]. For each instance, the heuristic proposed in Section 4 was utilized to obtain a feasible solution. We note here that for the heuristic, we choose to use the combination of the vehicle with the longest tour cost and the vehicle with the most number of targets for the maximal vehicle, i.e., to remove a target from. This is based on the results of our first parametric study, which is summarized in Fig. 6. Additionally, we perform a parametric study on the heuristic, wherein we compare the effects of the neighborhoods considered and the number of maximal vehicles considered on our final objective values, as well as the computation times. In particular, we compare the following variations:

- Neighborhood Type
  - One point move
  - One point move and One point swap
- Number of vehicles to consider from heuristic
  - One maximal vehicle based on the longest tour first, followed by a maximal vehicle based on the most number of targets.
  - Top two vehicles with the two highest longest tours, followed by the two vehicles with the most number of targets covered.

In addition, to demonstrate the efficacy of the heuristic on different α and τ values, four variations were considered for each instance, which are

- $\alpha = \frac{1}{TSP}, \tau = 1,$
- $\alpha = \frac{1}{TSP}, \tau = 2,$
- $\alpha = \frac{2}{TSP}, \tau = 1,$ and
- $\alpha = \frac{2}{TSP}, \tau = 2.$

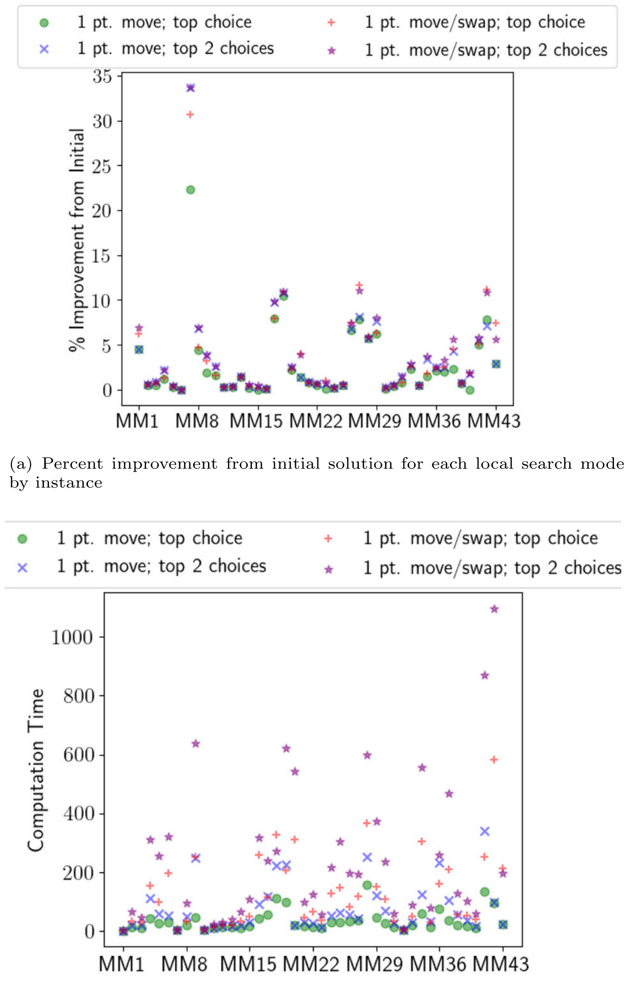
Here, *TSP* denotes the tour cost corresponding to a single vehicle covering all the targets, which is obtained using LKH.

For all these variations, the heuristic was run three times, and the best-obtained solution in terms of the objective value was selected. The minimum, maximum, mean, and median percentage improvement with respect to the initial solution over the 43 instances are reported in Table 4. Further, in this table, the minimum, maximum, mean, and median computation time over the 43 instances are reported. The percentage improvement and computation time for all the instances for $\alpha = \frac{1}{TSP}$, $\tau = 1$ for all variations in the heuristic are shown in Fig. 9. From the table and figure,

- For $\alpha = \frac{1}{TSP}$ and $\tau = 1$ or 2, a median improvement of around $1 - 2\%$, and a mean improvement of around $3 - 4\%$ was observed for all variations of the heuristic. Further, a maximum improvement of around 30% was observed. These observations emphasize the need for the local search and perturbation steps to improve the initial feasible solution obtained.
- Increasing α to $\frac{2}{TSP}$ leads to doubling the mean, median, and maximum improvements. Noting that α denotes a penalty factor that penalizes the revisit time, which in turn affects the dwell time, it can be observed that α influences the solution obtained for the same instance.
- The computation time for the heuristic for most cases is less than 10 minutes. Noting that the number of targets in the instances varies from 10 to 500, and the number of vehicles varies from 3 to 20, this observation indicates that the developed heuristic is practically applicable. The proposed heuristic was computationally efficient owing to the use of proxy costs to determine if an improved solution can be obtained from the incumbent (current

Table 4 Percentage improvement from initial solution and computation time by neighborhood search parameters

α	τ	Value	Method	Minimum	Median	Mean	Maximum
$\frac{1}{TSP}$	1	***Percentage Improvement***	1 pt. move; top choice	9.45e-06	1.43	3.05	29.7
			1 pt. move; top 2 choices	9.45e-06	1.91	3.69	34.04
			1 pt. move/swap; top choice	9.45e-06	1.56	3.42	30.68
			1 pt. move/swap; top 2 choices	9.45e-06	2.2	3.99	34.04
		Computation Time (s)	1 pt. move; top choice	1.78	31.85	45.88	171.93
			1 pt. move; top 2 choices	1.87	52.57	96.83	414.78
			1 pt. move/swap; top choice	4.58	88.34	140.87	504.57
			1 pt. move/swap; top 2 choices	2.31	165.29	274.21	1106.77
$\frac{1}{TSP}$	2	***Percentage Improvement***	1 pt. move; top choice	0.00e+00	1.21	2.98	29.76
			1 pt. move; top 2 choices	9.45e-06	2.20	3.67	34.08
			1 pt. move/swap; top choice	9.45e-06	1.41	3.36	30.73
			1 pt. move/swap; top 2 choices	9.45e-06	2.20	3.91	34.08
		Computation Time (s)	1 pt. move; top choice	1.82	29.99	40.56	145.35
			1 pt. move; top 2 choices	1.88	52.47	94.74	428.38
			1 pt. move/swap; top choice	3.94	86.93	144.46	604.44
			1 pt. move/swap; top 2 choices	2.30	140.59	281.96	1241.92
$\frac{2}{TSP}$	1	***Percentage Improvement***	1 pt. move; top choice	0.00e+00	2.83	6.38	68.73
			1 pt. move; top 2 choices	1.89e-05	3.85	7.75	79.8
			1 pt. move/swap; top choice	1.89e-05	3.16	7.12	70.61
			1 pt. move/swap; top 2 choices	1.89e-05	4.43	8.48	79.8
		Computation Time (s)	1 pt. move; top choice	2.12	30.54	43.04	178.28
			1 pt. move; top 2 choices	2.05	53.95	95.08	390.1
			1 pt. move/swap; top choice	2.05	79.0	137.84	551.94
			1 pt. move/swap; top 2 choices	2.51	153.99	272.83	972.27
$\frac{2}{TSP}$	2	***Percentage Improvement***	1 pt. move; top choice	0.00e+00	2.81	6.24	68.94
			1 pt. move; top 2 choices	1.89e-05	4.33	7.84	80.21
			1 pt. move/swap; top choice	1.89e-05	3.03	7.27	70.75
			1 pt. move/swap; top 2 choices	1.89e-05	4.46	8.67	80.21
		Computation Time (s)	1 pt. move; top choice	1.77	29.91	40.62	164.38
			1 pt. move; top 2 choices	1.99	61.6	105.12	436.97
			1 pt. move/swap; top choice	3.85	82.43	150.47	595.81
			1 pt. move/swap; top 2 choices	2.11	158.83	352.16	3588.55



(a) Percent improvement from initial solution for each local search mode by instance

(b) Compute times for each local search mode by instance

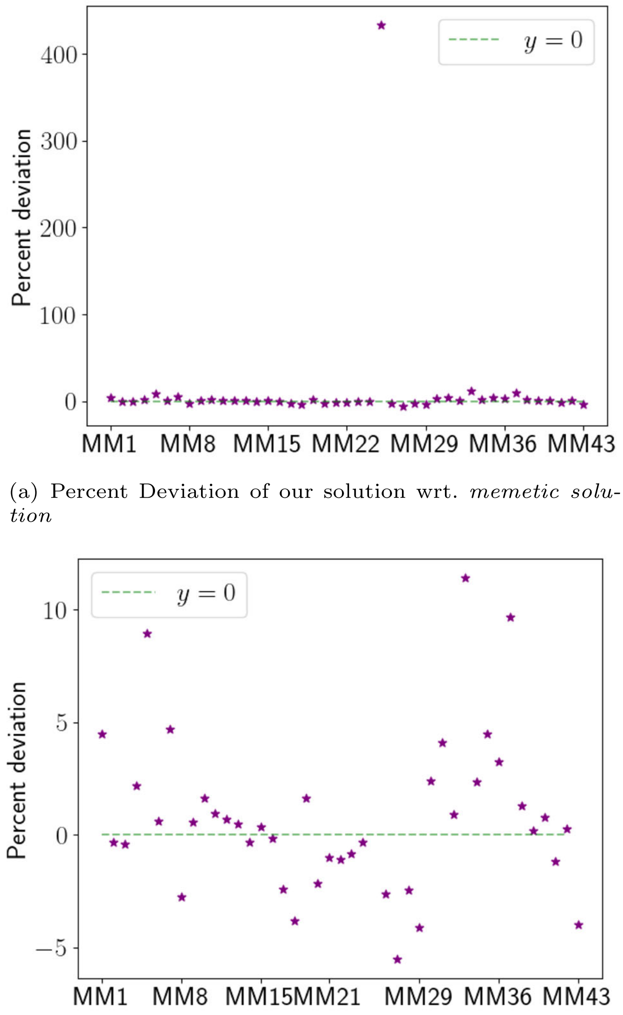
Fig. 9 Comparison between neighborhood search parameters with $\alpha = \frac{1}{TSP}$, $\tau = 1$

best) solution instead of optimizing each vehicle's tour and dwell time to evaluate every potential solution.

- Changing the neighborhood from 1 pt. move to 1 pt. move and swap leads to
  - A marginal change in the improvement in objective value. For example, for $\alpha = \frac{1}{TSP}$, $\tau = 1$, and top choice of maximal vehicle, the median improvement increased by 0.13%, the mean improvement increased by 0.37%, and the maximum improvement increased by 0.98%. Therefore, the inclusion of an additional neighborhood can be observed to marginally increase the quality of the solution obtained from the heuristic.
  - The computation time was about three times the previous computation time. Hence, the choice of utilizing 1 pt. move or 1 pt. move and swap depends on the user's preference: a higher quality solution, which corresponds to a higher computation time, or a good solution with a lower computation time.
- Changing the number of maximal vehicles considered from one vehicle to two vehicles leads to
  - A larger change in the improvement in objective value. For example, for $\alpha = \frac{1}{TSP}$, $\tau = 1$, and using 1 pt. move, the median improvement increased by 0.48%, the mean improvement increased by 0.64%, and the maximum improvement increased by 4.34%. The substantial improvement in the objective value is due to the higher number of candidate solutions explored in the neighborhoods.
  - The computation time was about twice the previous computation time. This observation is again due to the increased number of candidate solutions explored.

Remark Based on the presented results, it can be observed that using two vehicles for 1 pt. move or swap is recommended due to the higher-quality solutions obtained. Furthermore, based on the results, the choice of 1 pt. move with the top two vehicles yields the best trade-off between the objective value and computation time, whereas the choice of 1 pt. move and swap with the top two vehicles yields the best objective value.

In addition, we compare our algorithm with results using tours obtained from the memetic algorithm [39], the current best heuristic for the min-max multi-depot TSP. We utilize the partition of targets obtained from the memetic algorithm available at [40]. We compute the tour of each vehicle using LKH and the dwell time of targets covered by each vehicle using gradient descent. We choose to compare against the memetic algorithm since we expect the optimal solution to our problem to be similar to the min-max problem, as both problems require load balancing between vehicles. We expect this due to the results of Fig. 5, where we see that the objective value gained by a vehicle roughly approaches a maximum before the discounting term takes over. For $\alpha = \frac{1}{TSP}$ and $\tau = 2$, the objective values obtained using each method are shown in Fig. 10. For our heuristic, we used the 1 pt. move and swap, and the top two choices of vehicles. We observe that the percentage difference of our solution from the solution using the partition from the memetic algorithm had a mean percentage difference of +11% with a standard deviation of 65%, and a median difference of 0.35%. A depiction of solutions obtained using the heuristic and the memetic algorithm is shown for one of the instances in Fig. 11, wherein the heuristic yields an improved objective value. We can see that, on average, our algorithm performs marginally better than the solution obtained using the memetic algorithm. Further, we can see from Fig. 10 that there is one instance wherein our algorithm performs much better (up to a 400% improve-



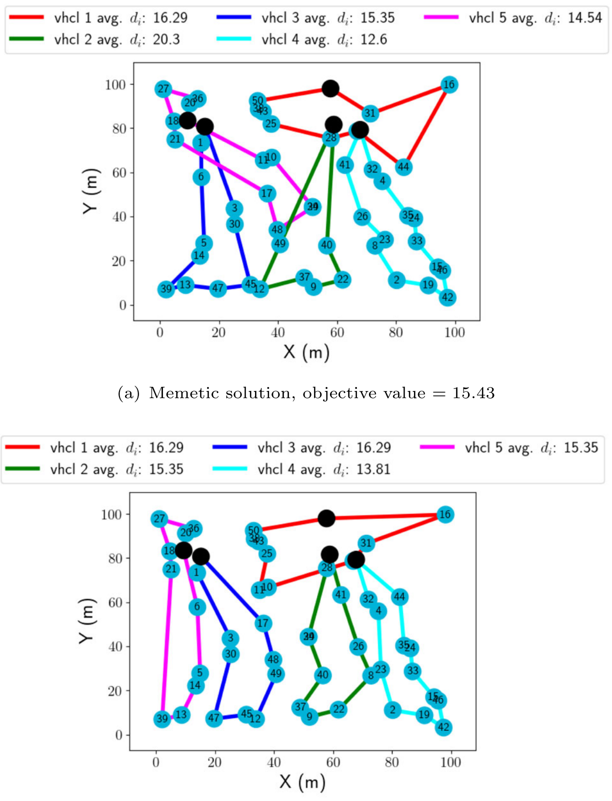
(a) Percent Deviation of our solution wrt. *memetic solution*

(b) Percent Deviation of our solution wrt. *memetic solution* (without outlier)

Fig. 10 Comparison of objective values from Min-Max tour solution vs. our algorithm using $\alpha = \frac{1}{TSP}$; $\tau = 2$. The percentage difference has range$(-5.5\%, 433.2\%)$ with a median of 0.35%. The mean and standard deviation are $(10.9\%, 65.3\%)$

ment), showing the benefit of our proposed heuristic for our proposed problem.

Through the comparison of our heuristic on standard instances with another state-of-the-art heuristic for a min-max problem, we have demonstrated the benefit of utilizing our proposed algorithm for the problem. We further demonstrate the performance of our algorithm on real-world examples. In particular, we consider six instances corresponding to maps depicting an urban region in different major cities in the world. We consider the center of each building in the maps to be a location to be visited/surveyed. The resulting tours obtained and the dwell time for the six instances are shown in Fig. 12.



(a) Memetic solution, objective value = 15.43

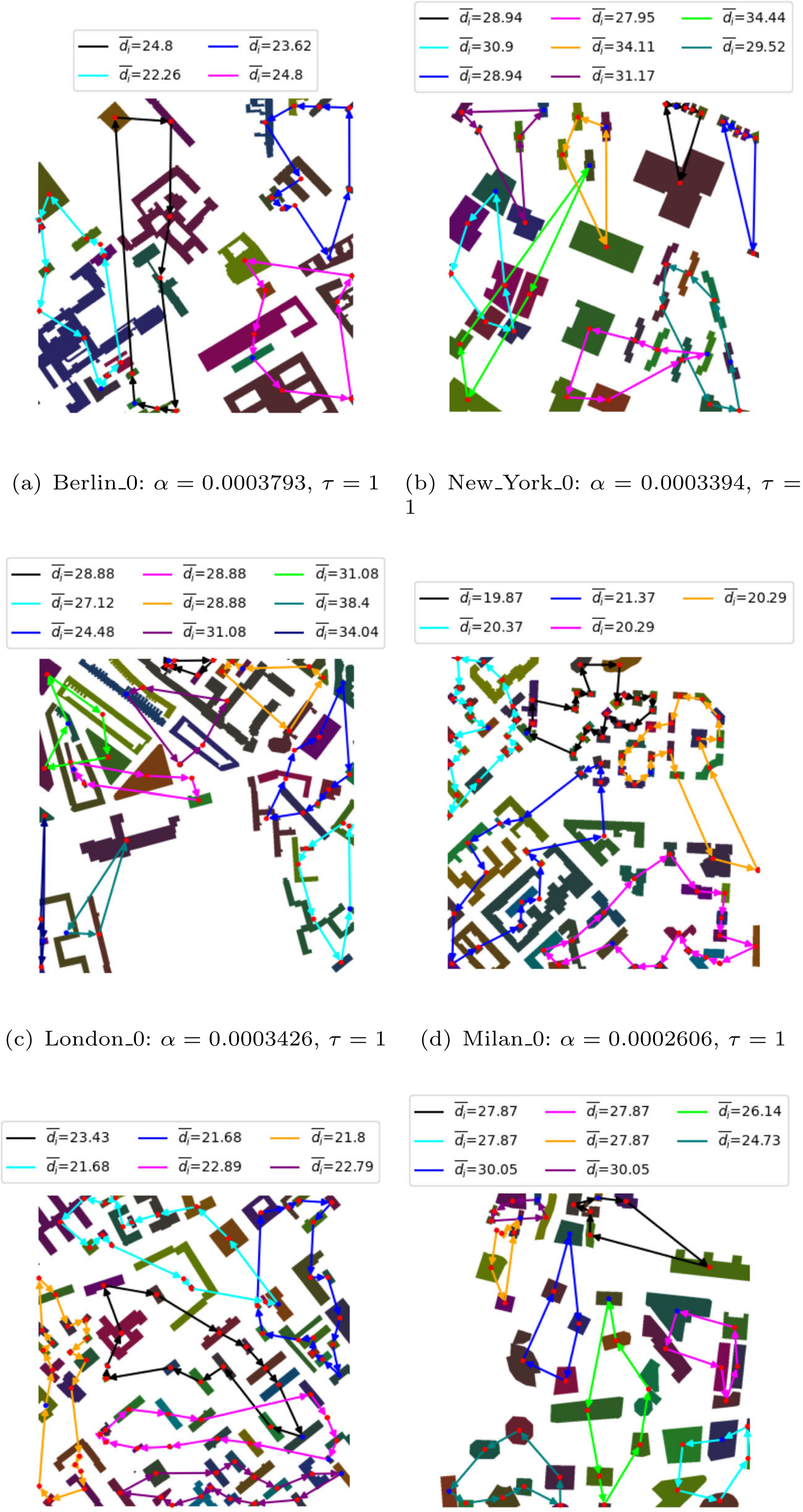
(b) Our solution, objective value = 15.69

Fig. 11 Comparison of MM10 Min-Max tour solution vs. our algorithm solution for $\alpha = \frac{1}{TSP} = 1.72 \cdot 10^{-3}$; $\tau = 2$

6 Conclusions & Remarks

In this study, a single and multi-UAV routing problem for enhancing the performance of a classifier-in-the-loop system was considered, wherein an operator provides points-of-interest (POIs) through an interface, and vehicles are required to collect information regarding the same. The information gained was mathematically formulated using Kullback-Leibler divergence and was discounted to ensure all POIs are visited. We considered two variants of the same problem: single-vehicle, and multi-vehicle. The single-vehicle problem was shown to be solved to optimality by decoupling the vehicle routing problem and optimizing the dwell time, which is the time spent at each POI. Numerical results for the same were presented to show that instances with 100 to 229 targets could be solved within 30 seconds. For

Fig. 12 Trials on real-world examples, obtained from https://movingai.com/benchmarks/street/index.html



(a) Berlin_0: $\alpha = 0.0003793$, $\tau = 1$

(b) New_York_0: $\alpha = 0.0003394$, $\tau = 1$

(c) London_0: $\alpha = 0.0003426$, $\tau = 1$

(d) Milan_0: $\alpha = 0.0002606$, $\tau = 1$

(e) Moscow_0: $\alpha = 0.0002486$, $\tau = 1$

(f) Sydney_0: $\alpha = 0.0003829$, $\tau = 1$

the multi-vehicle variant, a heuristic was proposed due to coupled partitioning and routing with dwell-time computation. Extensive numerical results were presented over varied types of instances, along with variations in the heuristic, to show that most of the instances and variations could be solved in 10 minutes. Furthermore, the heuristic was benchmarked with results obtained from a state-of-the-art heuristic for a min-max multi-depot traveling salesman problem. It was observed that our proposed heuristic yielded an improved solution by about 11% on average. Hence, the proposed heuristic produces high-quality solutions for the considered problem.

Acknowledgements Swaroop Darbha gratefully acknowledges the support of 2023 ONR Summer Faculty Fellowship Program.

Author Contributions
– Deepak Prakash Kumar: Implementation, Heuristic design, data analysis, manuscript writing
– Pranav Rajbhandari: Implementation, data analysis, manuscript writing
– Loy McGuire: Data analysis
– Swaroop Darbha: Study conception and design, manuscript writing
– Donald Sofge: Editing, problem formulation

Funding Swaroop Darbha gratefully acknowledges the support of 2023 ONR Summer Faculty Fellowship Program.

Declarations

Preprint submission: The initial draft of the paper is available on arXiv. The DOI is arXiv:2310.08828 and the URL is https://arxiv.org/abs/2310.08828, and the license of the preprint is arXiv.org perpetual, nonexclusive license.

References

1. Ji, S., Zhang, Z., Ying, S., Wang, L., Zhao, X., Gao, Y.: Kullback-leibler divergence metric learning. IEEE Transactions on Cybernetics. **52**(4), 2047–2058 (2022). https://doi.org/10.1109/TCYB.2020.3008248
2. Pandey, G., McBride, J.R., Savarese, S., Eustice, R.M.: Automatic extrinsic calibration of vision and lidar by maximizing mutual information. J. Field Robot. **32**(5), 696–722 (2015). https://doi.org/10.1002/rob.21542
3. Mladenović, N., Hansen, P.: Variable neighborhood search. Comput. Oper. Res. **24**(11), 1097–1100 (1997). https://doi.org/10.1016/S0305-0548(97)00031-2
4. Wang, X., Golden, B., Wasil, E.: The min-max multi-depot vehicle routing problem: heuristics and computational results. J. Oper. Res. Soc. **66**(9), 1430–1441 (2015). https://doi.org/10.1057/jors.2014.108
5. Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J.: The Traveling Salesman Problem: A Computational Study, Princeton University Press (2007)
6. Gutin, G., Punnen, A.P.: The Traveling Salesman Problem and Its Variations, Springer (2007)
7. Malik, W., Rathinam, S., Darbha, S.: An approximation algorithm for a symmetric generalized multiple depot, multiple travelling salesman problem. Oper. Res. Lett. **35**(6), 747–753 (2007). https://doi.org/10.1016/j.orl.2007.02.001
8. Oberlin, P., Rathinam, S., Darbha, S.: Today's traveling salesman problem. IEEE Robotics & Automation Magazine. **17**(4), 70–77 (2010). https://doi.org/10.1109/MRA.2010.938844
9. Oberlin, P., Rathinam, S., Darbha, S.: A transformation for a multiple depot, multiple traveling salesman problem. In: 2009 American Control Conference, pp. 2636–2641 (2009). https://doi.org/10.1109/ACC.2009.5160665
10. Helsgaun, K.: LKH TSP solver (2023). http://webhotel4.ruc.dk/~keld/research/LKH/
11. Rathinam, S., Sengupta, R., Darbha, S.: A resource allocation algorithm for multivehicle systems with motion constraints. IEEE Trans. Autom. Sci. Eng. **4**(1), 98–104 (2007)
12. Manyam, S.G., Rathinam, S., Darbha, S.: Computation of lower bounds for a multiple depot, multiple vehicle routing problem with motion constraints. Journal of Dynamic Systems, Measurement, and Control. **137**(9) (2015). https://doi.org/10.1115/1.4030354
13. Manyam, S.G., Rathinam, S., Casbeer, D., Garcia, E.: Tightly bounding the shortest dubins paths through a sequence of points. J. Intell. Robot. Syst. **88**, 495–511 (2017). https://doi.org/10.1007/s10846-016-0459-4
14. Dubins, L.E.: On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. Am. J. Math. **79** (1957). https://doi.org/10.2307/2372560
15. Hari, S.K.K., Rathinam, S., Darbha, S., Kalyanam, K., Manyam, S.G., Casbeer, D.: Optimal uav route planning for persistent monitoring missions. IEEE Trans. Rob. **37**(2), 550–566 (2021). https://doi.org/10.1109/TRO.2020.3032171
16. Hari, S.K.K., Rathinam, S., Darbha, S., Manyam, S.G., Kalyanam, K., Casbeer, D.: Bounds on optimal revisit times in persistent monitoring missions with a distinct and remote service station. IEEE Trans. Rob. (2022). https://doi.org/10.1109/TRO.2022.3210784
17. Hari, S.K.K., Rathinam, S., Darbha, S., Casbeer, D.W.: Cooperative Coverage with a Leader and a Wingmate in Communication-Constrained Environments (2022). https://doi.org/10.48550/arXiv.2210.02628
18. Bhadoriya, A.S., Montez, C.M., Rathinam, S., Darbha, S., Casbeer, D.W., Manyam, S.G.: Optimal path planning for a convoy-support vehicle pair through a repairable network. IEEE Trans. Autom. Sci. Eng. 1–12 (2023) https://doi.org/10.1109/TASE.2023.3305392
19. Kalyanam, K., Pachter, M., Patzek, M., Rothwell, C., Darbha, S.: Optimal human-machine teaming for a sequential inspection operation. IEEE Trans. Hum.-Mach. Syst. **46**(4), 557–568 (2016). https://doi.org/10.1109/THMS.2016.2519603
20. Cover, T.M., Thomas, J.A.: Elements of Information Theory, Wiley (2006)
21. Lee, C.-Y., Lee, Z.-J., Lin, S.-W., Ying, K.-C.: An enhanced ant colony optimization (eaco) applied to capacitated vehicle routing problem. Appl. Intell. **32**, 88–95 (2010). https://doi.org/10.1007/s10489-008-0136-9

22. Du Toit, N.E., Burdick, J.W.: Robot motion planning in dynamic, uncertain environments. IEEE Trans. Rob. **28**(1), 101–115 (2011). https://doi.org/10.1109/TRO.2011.2166435
23. Kaufman, E., Lee, T., Ai, Z.: Autonomous exploration by expected information gain from probabilistic occupancy grid mapping. In: 2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), IEEE, pp. 246–251 (2016). https://doi.org/10.1109/SIMPAR.2016.7862403
24. Zaenker, T., Rückin, J., Menon, R., Popović, M., Bennewitz, M.: Graph-based view motion planning for fruit detection. arXiv preprint arXiv:2303.03048. (2023) https://doi.org/10.48550/arXiv.2303.03048
25. Paull, L., Saeedi, S., Li, H., Myers, V.: An information gain based adaptive path planning method for an autonomous underwater vehicle using sidescan sonar. In: 2010 IEEE International Conference on Automation Science and Engineering, IEEE, pp. 835–840 (2010). https://doi.org/10.1109/COASE.2010.5584478
26. Mostofi, Y.: Decentralized communication-aware motion planning in mobile networks: an information-gain approach. J. Intell. Rob. Syst. **56**, 233–256 (2009). https://doi.org/10.1007/s10846-009-9335-9
27. Novakovic, J.: Using information gain attribute evaluation to classify sonar targets. In: 17th Telecommunications Forum TELFOR, pp. 1351–1354 (2009). Citeseer
28. Deng, D., Duan, R., Liu, J., Sheng, K., Shimada, K.: Robotic exploration of unknown 2d environment using a frontier-based automatic-differentiable information gain measure. In: 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), IEEE, pp. 1497–1503 (2020). https://doi.org/10.1109/AIM43001.2020.9158881
29. Julian, B.J., Karaman, S., Rus, D.: On mutual information-based control of range sensing robots for mapping applications. The International Journal of Robotics Research. **33**(10), 1375–1392 (2014). https://doi.org/10.1177/0278364914526288
30. Bai, S., Wang, J., Chen, F., Englot, B.: Information-theoretic exploration with bayesian optimization. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 1816–1822 (2016). https://doi.org/10.1109/IROS.2016.7759289
31. Amigoni, F., Caglioti, V.: An information-based exploration strategy for environment mapping with mobile robots. Robot. Auton. Syst. **58**(5), 684–699 (2010). https://doi.org/10.1016/j.robot.2009.11.005
32. Basilico, N., Amigoni, F.: Exploration strategies based on multi-criteria decision making for searching environments in rescue operations. Auton. Robot. **31**, 401–417 (2011). https://doi.org/10.1007/s10514-011-9249-9
33. Alipour, M., Faez, K.: On design of mobile agent routing algorithm for information gain maximization in wireless sensor networks. In: International Conference on Systems and Networks Communications (2011)
34. Montez, C., Darbha, S., Valicka, C., Staid, A.: Routing of an unmanned vehicle for classification. In: Pham, T., Solomon, L., Rainey, K. (eds.) Artificial Intelligence And Machine Learning For Multi-domain Operations Applications II, vol. 11413, p. 1141319 (2020). https://doi.org/10.1117/12.2558748
35. Kumar, S.: On maximizing the total information gain in a vehicle routing problem. Master's thesis, Texas A&M University (2023)
36. Carlsson, J., Ge, D., Subramaniam, A., Wu, A., Ye, Y.: Min-max multi-depot vehicle routing problem. Lect. Global Optim. **6**(1), 141–152 (2012)
37. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2023). https://www.gurobi.com
38. Reinelt, G.: **TSPLIB**. Accessed 22 May 2023 (1995). http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/
39. He, P., Hao, J.-K.: Memetic search for the minmax multiple traveling salesman problem with single and multiple depots. Eur. J. Oper. Res. **307**(3), 1055–1070 (2023). https://doi.org/10.1016/j.ejor.2022.11.010
40. He, P. Accessed Sept 2023. https://github.com/pengfeihe-angers/minmax-mTSP

Deepak Prakash Kumar is a Ph.D. candidate in the Department of Mechanical Engineering at Texas A&M University. He holds a B.Tech in Engineering Design and an M.Tech in Automotive Engineering from the Indian Institute of Technology Madras. His research interests encompass path planning for unmanned aerial vehicles, algorithms for vehicle routing problems, and vehicle dynamics and control.

Pranav Rajbhandari is currently pursuing their BS in Artificial Intelligence and Discrete Mathematics at the School of Computer Science at Carnegie Mellon University. Their research areas of interest include reinforcement learning algorithms, sim-to-real transfer, topology, and extremal combinatorics.

Loy McGuire received the B.S. degree in mechanical engineering, manufacturing engineering, and energy from Miami University, Oxford, OH, USA in 2017, and in 2020 he received the M.S. degree in aerospace engineering from the University of Maryland, College Park, MD, USA, where he is currently pursuing the PhD degree in aerospace engineering. Since 2018 he has worked as an intern at the Naval Research Laboratory within the Distributed Autonomous Systems Section in the Navy Center for Applied Research in Artificial Intelligence (NCARAI), where he currently researches autonomous multi-agent and swarm robotic path planning algorithms. He has also been a member of the Motion and Teaming Laboratory at the University of Maryland since 2018.

Swaroop Darbha received the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA, in 1994. He is currently a Professor in Mechanical Engineering with Texas A&M University, College Station, TX, USA. He is a Fellow of ASME and IEEE for his contributions to Intelligent Transportation Systems and Unmanned Vehicles. Dr. Darbha's current research interests include Aiding Human-In-the-Loop Classification with UAS, Infrastructure & Resource assisted Routing of UAVs, Energy-Aware Autonomy, Planning for Advanced Air Mobility, and 3-D Motion Planning of UAVs.

Donald Sofge is a Roboticist and Section Head at the U.S. Naval Research Laboratory (NRL) with 33 years of experience (23 at NRL) in Artificial Intelligence and Control Systems R&D. He leads the Distributed Autonomous Systems Section in the Navy Center for Applied Research in Artificial Intelligence (NCARAI), where he develops nature-inspired computing paradigms to challenging problems in sensing, artificial intelligence, and control of autonomous robotic systems. His current research focuses on control of autonomous teams or swarms of robotic systems for Navy relevant missions. He has served as PI on dozens of federally funded R&D programs, and has more than 200 peer-reviewed publications (including 11 books) on autonomy, intelligent control, quantum computing, and related topics. He has served as an advisor on autonomous systems to DARPA, ONR, OSD, ARL, NSF, and NASA, as well as US representative on international TTCP and NATO technical panels on autonomous systems, and has participated as a member of the National Science and Technology Council (NSTC) Networking and Information Technology Research and Development (NITRD) Program Interagency Working Groups: Intelligent Robotics and Autonomous Systems (IRAS), Machine Learning and Artificial Intelligence (MLAI), and the AI R&D Ad Hoc Group. Don also serves on the Academic Advisory Board for the Maryland Robotics Center (MRC) at the University of Maryland and occasionally serves as an Adjunct Faculty Member teaching graduate-level courses in Robotics.