This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Optimal UAV Route Planning for Persistent Monitoring Missions

Sai Krishna Kanth Hari, Sivakumar Rathinam, *Senior Member, IEEE*, Swaroop Darbha, *Fellow, IEEE*, Krishna Kalyanam, *Senior Member, IEEE*, Satyanarayana Gupta Manyam, and David Casbeer, *Senior Member, IEEE*

***Abstract*—This article addresses a persistent monitoring problem (PMP) that requires an unmanned aerial vehicle (UAV) to repeatedly visit n targets of equal priority. The UAV has limited onboard fuel/charge and must be regularly serviced at a depot. Given a fixed number of visits, k, for the UAV to the targets between successive services, the objective of the PMP is to determine an optimal sequence of visits such that the maximum time elapsed between successive visits to any target is minimized. This planning problem is a generalization of the traveling salesman problem and is NP-hard. We characterize the optimal solutions to this problem for different values of k and develop algorithms that can compute the optimal solutions relatively fast. Numerical results are also presented to corroborate the performance of the proposed approach.**

***Index Terms*—Motion and path planning, optimization and optimal control, surveillance systems, sensor networks, unmanned vehicles.**

I. Introduction

PERSISTENT monitoring missions are repetitive in nature and last for prolonged periods of time. In this work, we consider an optimal route planning problem arising in persistent monitoring missions that require a UAV to repeatedly visit n targets of which one target (referred to as the depot) also acts as a recharging or a refueling station. For the optimal performance of these missions, it is desirable to make the revisit time, defined as the maximum of the time elapsed between successive visits to any target, as small as possible.

Manuscript received March 10, 2020; revised July 8, 2020; accepted August 25, 2020. This work was supported by the Air Force Research Laboratory. This article was recommended for publication by Associate Editor J. O'Kane and Editor P. Robuffo Giordano upon evaluation of the reviewers' comments. *(Corresponding author: Sai Krishna Kanth Hari.)*

Sai Krishna Kanth Hari is with the Applied Mathematics and Plasma Physics Division, Los Alamos National Laboratory, Los Alamos, NM 87544 USA (e-mail: hskkanth@gmail.com).

Sivakumar Rathinam and Swaroop Darbha are with the Department of Mechanical Engineering, Texas A&M University Texas, College Station, TX 77843 USA (e-mail: srathinam@gmail.com; dswaroop@tamu.edu).

Krishna Kalyanam is with the Palo Alto Research Center, Palo Alto, CA 94304 USA (e-mail: krishnak@ucla.edu).

Satyanarayana Gupta Manyam is with the Infoscitex corporation, Dayton, OH 45431 USA (e-mail: msngupta@gmail.com).

David Casbeer is with the Autonomous Control Branch, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH 45433 USA (e-mail: david.casbeer@afresearchlab.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2020.3032171

Due to a limited onboard fuel/charge capacity, the UAV must be regularly serviced (recharged) at the depot, and each service requires Δ time units.

To ensure a safe return to the depot for servicing, the maximum operational range of the UAV must be estimated. As noted in [1], estimating the operational range provided to a UAV by its battery (or any other fuel source) is nontrivial due to several reasons: dependence of the battery charge consumption on unanticipated weather conditions, degradation of battery capacity over discharge cycles, losses in energy due to internal friction, usage of the battery charge for other ancillary functions such as communication between sensors and the environment and on-board computations. Therefore, we choose a simple model, which *conservatively* estimates the operational range in terms of the number of visits made by the UAV in a servicing cycle. Estimating the operational range in terms of the number of visits requires caution as visits between different targets could require the UAV to travel different distances. Therefore, in order to ensure that the model is suitable for arbitrary target configurations, we propose an estimation procedure (see Appendix A) that takes into account the relative positioning of the targets and the actual sequence of visits made by the UAV;[1] the conservative estimate of the operational range is denoted by $\bar{k}$ visits. With this model, the addressed problem, referred to as the persistent monitoring problem (PMP), is defined as follows:

Given that the UAV is allowed to make exactly k visits between any two consecutive service stops to the depot, find a sequence of k visits with the least revisit time.

To simplify the algorithm development and analysis, we assume that $k \geq n$ and that each target must be visited at least once between any two service stops for the UAV.[2] A smaller value of k implies that the UAV must return to the depot more frequently for recharging or refueling, which increases the total service time of the UAV over a longer time period. This article will provide a systematic way to address the PMP for any $k \in [n, \bar{k}]$ so that

[1] A working example of the estimation procedure is provided in Section VI-B3.

[2] Without this assumption, the PMP is a generalization of the orienteering problem, which is much harder to theoretically analyze and to solve.

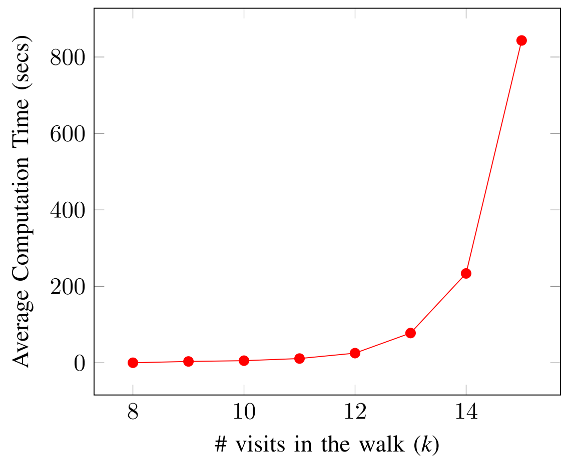


Fig. 1. Plot showing the rapid increase in the average computation time for solving the PMP using the MBLP against the number of visits in the walk; the average was taken over 50 instances with eight targets. The simulations were performed on a MacBook Pro with 16 GB RAM, Inter Core i7 processor and a processor speed of 2.5 GHz. In these instances, we assume $\Delta = 0$.

the tradeoffs between the revisit times and the service times can be analyzed.[3]

Unlike commonly addressed cost functions such as the total fuel cost or travel time, modeling, and solving for the minimum revisit time is relatively harder as one needs to keep track of the times elapsed between visits to each target as well as decide the sequence in which the targets must be visited. If $k = n$, each target is visited exactly once between any two consecutive visits to the depot; therefore, the revisit time for any target equals the sum of all the travel times involved in visiting all the targets in a given sequence. Hence, the PMP reduces to the classic traveling salesman problem (TSP) for $k = n$, and is, therefore, NP-hard. To illustrate the difficulty in solving the PMP, particularly when $k > n$, see Fig. 1, which shows the plot of the average computation time required to solve a mixed binary linear programming (MBLP) formulation of the PMP [2] as a function of the number of visits for just eight targets. For instances with larger number of targets, the MBLP formulation was interminable; for example, an instance with $n = 20$ and $k = 21$ required 165 273 s on a standard laptop rendering the MBLP formulation impractical.

The main contributions of this article are as follows.

1) *When the service time is negligible, i.e.,* $\Delta \approx 0$: We show that it is sufficient to find the optimal solutions for the PMP when $k \in \{n, n+1, \ldots, 2n-1\}$. These n optimal solutions are relatively easy to compute using standard solvers. We show these solutions can then be easily transformed to optimal solutions for any $k \geq 2n$. Furthermore, we show that it is sufficient to compute an optimal solution for either n or $n+1$ visits to determine an optimal solution for any $k \geq n^2 - n$.
2) *When the servicing time is relatively large compared to the travel times between the targets,* $i.e.$, $\Delta \geq 2c_{\min}$ *where* $c_{\min}$ *is the smallest travel time between any two targets*: We show that it is sufficient to compute an optimal TSP tour over the targets to determine the optimal solution for all $k \geq n^2 + n$.
3) *When the servicing time is any positive real number,* $i.e.$, $\Delta > 0$: We construct feasible solutions for the PMP that are at most Δ units away from the optimum.

In addition, we also provide extensive numerical results to corroborate the performance of the methods proposed to efficiently solve the PMP.

[3]Even if the service time is negligible, there is an incentive in making more than n visits. Suppose the UAV is powered by disposable batteries. Then, replacing the batteries prior to consuming a majority of its charge leads to a wastage of the unused energy. In such a case, by solving the PMP for different values of k, one can perform a tradeoff analysis between the revisit time and the wasted energy.

II. Literature Review

Route planning in persistent monitoring missions is different from other search or exploration missions due to the requirement of repeatedly visiting certain targets in the environment. With the aim of obtaining an up-to-date knowledge of the changing information, the problem of finding routes with the least revisit time was first proposed in [3]. Since then, several researchers have worked on the problem of finding an optimal sequence of visits for persistent monitoring of targets (with equal or unequal priorities) using single or multiple UAVs [4]–[13]. However, the solution methods are either heuristics or approximation algorithms due to the computational complexity of the problem [14]. Furthermore, most of the prior works do not account for the limited fuel/charge capacity of the UAV and the time required to service the UAV. As highlighted in [1] and [15], accounting for these factors is an important part of the planning problem and neglecting the same can lead to routes with significantly higher revisit times in practice than those planned for. The overall time spent on the servicing the UAV can also be larger than that for the optimal case. For this reason, it is imperative to first focus on a *single UAV* and find optimal routes (routes with the least revisit time) corresponding to different service schedules (different values of k) and different amounts of service time Δ. This helps in designing routes that yield lower revisit time and total service time in practice. Solution procedures developed for this problem can also be used as subroutines to solve the general problem of planning routes for the multiple UAV case and its variants [16] such as finding the minimum number of UAVs required for monitoring.

In [17], the authors consider a single UAV PMP for targets with arbitrary weights/priorities. Here, the objective is to minimize the weighted revisit time, which is defined as the maximum among the products of the time between successive visits to the targets and the weights of the corresponding targets. This work considers walks with infinite time horizon, without accounting for the finite operational range of the UAV or the time required to service the UAV at the depot. The authors show that optimal infinite walks can be obtained by indefinitely repeating finite walks.

In our work, we consider finite walks, accounting for the limited operational range of UAV. The final visit of a walk is back to the depot for servicing the UAV. Once the UAV is serviced, the same walk is repeated until the end of the

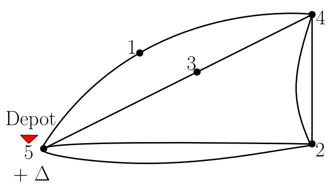


Fig. 2. Graphical representation of a walk $\mathcal{W}_\Delta(8) = (5, 1, 4, 2, 5, 3, 4, 2, 5)$ with eight visits.

mission. While one can also consider taking a different walk after each service, such a modeling decision involves a number of challenges. First, one must plan the set of all walks the UAV takes in order to minimize the amortized revisit time. In several monitoring missions, the total mission duration is unknown. In this case, one does not know how long a walk to plan. Even if the mission duration is known *a priori*, planning such walks involves a number of modeling and computational challenges (see Appendix C); the limited time available for planning walks prior to the execution of mission does not necessarily permit such expensive computations. Nonetheless, in Section IV-D, we discuss cases in which one cannot obtain a lower revisit time by taking different walks after each service, i.e., repeating the same walk is optimal.

Furthermore, in [17], the authors show that the number of visits in a finite walk that needs to be repeated to obtain an optimal infinite walk can be exponentially large. In this work, for the case of equally weighted targets, we show that an optimal walk with $k \geq 2n$ visits can be constructed by systematically repeating an optimal walk with at most $2n-1$ visits (see Section IV). Furthermore, when $k \geq n^2 - n$, we show that it is sufficient to compute and systematically repeat an optimal walk with either n or $n+1$ visits (refer to the remarks following Corollary 4).

III. Terminology and Notation

Let the set of all targets be denoted by the set $\mathcal{T} = \{1, 2, \ldots, n\}$. The travel times between any two distinct targets $u, v\ (u \neq v)$ in $\mathcal{T}$ be denoted by $c(u, v) > 0$. We assume that the targets are all of equal priority, and the travel times between them satisfy the triangle inequality, i.e., $c(u, v) + c(v, w) \geq c(u, w)$ for all $u, v, w \in \mathcal{T}$.

A. Definitions

Walk: A walk denotes the sequence in which the UAV visits the targets. Given that the UAV is serviced after every k visits and each service requires Δ units of time, a walk with k visits and a service time of Δ is denoted by $\mathcal{W}_\Delta(k) = (v_0, v_1, \ldots, v_k)$, where $v_i \in \mathcal{T}$ and $v_i \neq v_{i+1}$ for $i = 0, \ldots, k-1$, and $v_0 = v_k$. Note the following:

1) the first element v_0 in the sequence indicates the depot and is associated with a service time of Δ units;
2) the first visit in the walk is to v_1, second visit is to v_2 and so on;
3) each visit must be between two distinct targets.

An example walk with eight visits to five targets and a service stop of Δ time units at the depot (target 5) is shown in Fig. 2.

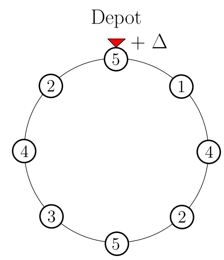


Fig. 3. Cyclic representation of the walk $\mathcal{W}_\Delta(8) = (5, 1, 4, 2, 5, 3, 4, 2, 5)$ (shown in Fig. 2) with node 5 as the depot. Example revisit sequences of this walk are $(1, 4, 2, 5, 3, 4, 2, 5, 1)$, $(2, 5, 3, 4, 2)$, $(2, 5, 1, 4, 2)$.

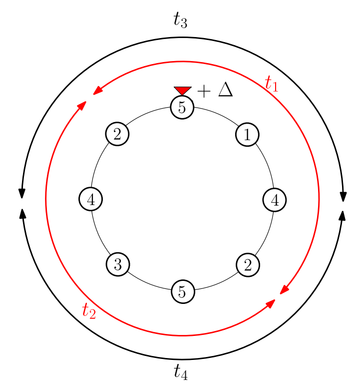


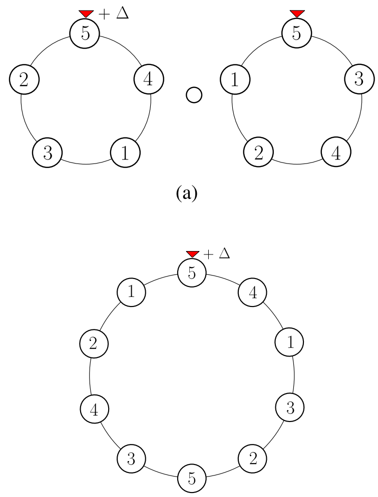
Fig. 4. Revisit times of targets 2 (t_1 and t_2) and 4 (t_3 and t_4) in the walk $\mathcal{W}(8)$ considered in Fig. 2.

Since a walk is repeated indefinitely, we depict it using a cyclic representation, as shown in Fig, 3. The total time required to traverse through all the nodes in a walk is referred to as the *travel time* of the walk.

Revisit sequence: A contiguous subsequence of a walk in which the first element is the same as the last element, but is different from any of the intermediate elements is referred to as a revisit sequence of the walk. Refer to Fig. 3, for examples. Note that the time taken to traverse through a revisit sequence is the time between successive visits to the target that is the terminal element of the sequence. *So, by definition, the revisit time of a walk is the maximum travel time over all the revisit sequences in the walk.*

Fig. 4 illustrates the time between successive visits to targets 2 and 4 in the walk $\mathcal{W}(8)$. It is to be noted that the revisit sequences that include a service stop to the depot have an additional travel time of Δ units. That is, in Fig. 4, we have $t_1 = c(2, 5) + \Delta + c(5, 1) + c(1, 4) + c(4, 2)$.

Concatenation: The concatenation of two walk sequences: $\mathcal{AW}_\Delta = (d, v_1, \ldots, v_k, d)$ and $\mathcal{BW}_0 = (d, u_1, \ldots, u_l, d)$ is defined as $\mathcal{AW}_\Delta \circ \mathcal{BW}_0 := (d, v_1, \ldots, v_k, d, u_1, \ldots, u_l, d)$. Note that only the first walk in the concatenation can have a nonzero service time. In other words, the UAV is serviced at the depot only after completing all the visits specified by a concatenated walk and the service time associated with the concatenated walk is also exactly Δ units. Fig. 5(b) shows a walk obtained by concatenating the two walks in Fig. 5(a). The concept of concatenation is helpful in the construction of feasible walks with larger number of visits.



(a)

(b)

Fig. 5. Concatenation of two walks with five visits each to forma walk with ten visits. (a) Two walks, each with five visits, are concatenated. (b) Result of the concatenation [in (a)] is a walk with ten visits.

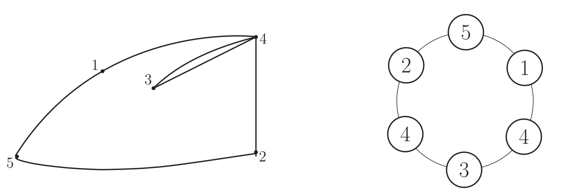


Fig. 6. Shortcut walk $\mathcal{SW}_\Delta(6)$ obtained by shortcutting visits to targets 2 and 5 from the walk $\mathcal{W}_\Delta(8) = (5, 1, 4, 2, 5, 3, 4, 2, 5)$.

Hereafter, for convenience of representation, the depot and its servicing time are not indicated in the cyclic representation of a walk.

Shortcut walk: Given a walk, $\mathcal{W}_\Delta = (d, \ldots, v, \ldots d)$, skipping (removal of) a visit v from $\mathcal{W}_\Delta$ is referred to as shortcutting the visit v from $\mathcal{W}_\Delta$. For a walk $\mathcal{SW}_\Delta$ to be referred to as a shortcut walk of $\mathcal{W}_\Delta$: 1) it must be possible to form $\mathcal{SW}_\Delta$ by shortcutting one or more visits from $\mathcal{W}_\Delta$; 2) none of the visits that are shortcut must be the last visit to a target in $\mathcal{W}_\Delta$. As an illustration, consider the walk $\mathcal{W}_\Delta(8) = (5, 1, 4, 2, 5, 3, 4, 2, 5)$ shown in Fig. 2. $\mathcal{W}_\Delta(8)$ has two visits to each of targets 2, 4, and 5, and one visit to each of targets 1 and 3. So, its shortcut walk can be formed by shortcutting only the first visits to targets 2, 4, and 5. For example, (5,1,4,3,4,2,5), which is shown in Fig. 6, is a shortcut walk of $\mathcal{W}_\Delta(8)$ obtained by shortcutting the first visits to targets 2 and 5, but retaining the rest of the visits in the order they appear in $\mathcal{W}_\Delta(8) = (5, 1, 4, 2, 5, 3, 4, 2, 5)$. Note that a shortcut walk must satisfy all the requirements of a walk.

The abovementioned definitions are useful in proving the results presented in the subsequent sections. We will also use the following additional notations.

1) $\mathcal{R}(\mathcal{W}_\Delta(k))$—revisit time of a walk $\mathcal{W}_\Delta(k)$ (including a service time of Δ).
2) $T(S)$—travel time of a sequence S_Δ (including a service time of Δ).
3) $\mathcal{R}^*_\Delta(k)$—optimal revisit time for k visits when the service time is Δ.
4) $T^*_\Delta(k)$—minimum travel time possible for a walk with k visits when the service time is Δ.
5) $\mathcal{W}^*_\Delta(k)$—optimal walk with k visits when the service time is Δ.

Using the abovementioned notations, the key results shown in this article are summarized in Table I. For various values of Δ and k, the table lists conditions on the optimal revisit times, $\mathcal{R}^*_\Delta(k)$ (column 3), and walks with the smallest number of visits that need to be computed to determine an optimal walk, $\mathcal{W}^*_\Delta(k)$, using the procedure described in this article (column 4).

IV. Solution to the PMP With Negligible Service Time

In this section, we characterize optimal solutions for the case when the service time is negligible in comparison with the travel times. Specifically, we set $\Delta = 0$. This characterization is expressed mathematically through the following main theorem.

Theorem 1: Given a number of visits k such that $k \geq n$, it can be expressed as $k = pn + q$ where $p, q \in Z_+$, $p \geq 1$, and $q \leq n - 1$. Then, $\mathcal{R}^*_0(k) = \mathcal{R}^*_0(n + \lceil \frac{q}{p} \rceil)$.

Theorem 1 will be proved using several Lemmas in the remainder of this section. Specifically, Lemmas 1–3 show some key properties regarding concatenation and revisit sequences of walks, and Lemmas 4–6 are established to prove the bounds on the optimal revisit times. All these Lemmas are then leveraged to prove Theorem 1 in Section IV-C. We then use this theorem to develop efficient formulations for solving the problem, making an online computation of optimal routes viable.

Lemma 1: Concatenating a walk, $\mathcal{W}_0(k)$, with itself any number of times does not increase its revisit time.

Proof: Let $\mathcal{RV}$ be the set of all revisit sequences of $\mathcal{W}_0(k)$. Now, construct a walk $\mathcal{CW}_0(mk)$ by concatenating $\mathcal{W}_0(k)$ with itself m times. Then, it is easy to see that the set of all revisit sequences of $\mathcal{CW}_0(mk)$ is also $\mathcal{R}V$. Since the revisit time of a walk is defined as the maximum among the travel times of all the revisit sequences of the walk, both $\mathcal{W}_0(k)$ and the concatenated walk, $\mathcal{CW}_0(mk)$, have the same revisit time. ■

Lemma 2: If $n + 1 \leq k \leq 2n - 1$, concatenating a walk, $\mathcal{W}_0(k)$, with itself or its shortcut walk with $k - 1$ visits, $\mathcal{SW}_0(k-1)$, any number of times and in any order, does not increase its revisit time.

Proof: Let $p, q \geq 0$ be two integers that are not simultaneously zero. Let $\mathcal{CSW}_0(pk + q(k-1))$ be obtained by concatenating p instances of $\mathcal{W}_0(k)$ and q instances of $\mathcal{SW}_0(k-1)$ in any specified order. One can think of $\mathcal{CSW}_0$ as a shortcut walk of $\mathcal{CW}_0((p+q)k)$ that is obtained by concatenating $\mathcal{W}_0(k)$ with itself $(p + q)$ times. It is easy to see that the revisit sequences of $\mathcal{CSW}_0(k)$ are either the revisit sequences of $\mathcal{W}_0(k)$ or the shortcut versions of the revisit sequences in $\mathcal{W}_0(k)$. By triangle inequality, the travel times of shortcut versions of a sequence cannot increase the travel time of the sequence itself. Hence, the revisit time of $\mathcal{CSW}_0$ cannot exceed that of $\mathcal{W}_0(k)$.

Since the set of revisit sequences of $\mathcal{W}_0(k)$ is a subset of that of $\mathcal{CSW}_0(k)$, it also follows that the revisit time of $\mathcal{W}_0(k)$

TABLE I
SUMMARY OF KEY RESULTS PROVED IN SECTIONS IV AND V

Δ	k	$\mathcal{R}^*_\Delta(k)$	$\mathcal{W}^*_\Delta(k)$
$\Delta = 0$	k is an integer multiple of n	$\mathcal{R}^*_0(k) = \mathcal{R}^*_0(n)$	It is sufficient to compute $\mathcal{W}^*_0(n)$.
	$k \geq n^2 - n$	$\mathcal{R}^*_0(k) = \begin{cases} \mathcal{R}^*_0(n), \text{if } k \text{ is an integer multiple of } n; \\ \mathcal{R}^*_0(n+1), \text{ otherwise.} \end{cases}$	It is sufficient to compute either $\mathcal{W}^*_0(n+1)$ or $\mathcal{W}^*_0(n)$.
	$k \geq 2n$	$\mathcal{R}^*_0(k) \subset \{\mathcal{R}^*_0(n), \mathcal{R}^*_0(n+1), \ldots, \mathcal{R}^*_0(2n-1)\}$; $\mathcal{R}^*_0(k) = \mathcal{R}^*_0(n + \left\lceil \frac{q}{p} \right\rceil)$, where $k = pn + q$, $p, q \in Z_+$, $p \geq 1$, and $q \leq n-1$.	It is sufficient to compute $\mathcal{W}^*_0(n + \lceil \frac{q}{p} \rceil)$, i.e., one among the set: $\{\mathcal{W}^*_0(n), \mathcal{W}^*_0(n+1), \ldots, \mathcal{W}^*_0(2n-1)\}$.
	$n \leq k \leq 2n-1$	$\mathcal{R}^*_0(k)$ is a non-decreasing function of k	$\mathcal{W}^*_0(k)$ can be used to construct optimal walks with more than $2n-1$ visits.
$\Delta > 0$ (includes $\Delta \geq 2c_{min}$)	$n \leq k \leq 2n-1$	$\mathcal{R}^*_\Delta(k) = \mathcal{R}^*_0(k) + \Delta$	$\mathcal{W}^*_0(k)$ is an optimal walk.
	$k \geq 2n$	$\mathcal{R}^*_\Delta(k) \leq \mathcal{R}^*_0(k) + \Delta$	$\mathcal{W}^*_0(k)$ is a feasible walk.
$\Delta \geq 2c_{min}$	$k \geq n^2 + n$	$\mathcal{R}^*_\Delta(k) = \mathcal{R}^*_\Delta(n)$	It is sufficient to compute $\mathcal{W}^*_0(n)$.

cannot exceed that of $\mathcal{CSW}_0(k)$. Combining, we get that the revisit time of $\mathcal{W}_0$ equals that of $\mathcal{CSW}_0$. ∎

Lemma 3: A walk with $k = pn + q$ visits, where $p, q \in Z_+$, $p \geq 1$, and $0 \leq q \leq n-1$, has a revisit sequence with at least $n + \lceil \frac{q}{p} \rceil$ visits.

Proof: A walk with $pn + q$ visits must have a target that is visited at least p times; otherwise, if every target is visited less than p times, then the total number of visits in the walk is less than pn, which is a contradiction to the fact that the walk has $pn + q$ visits with $q \geq 0$.

Consider such a target (that is visited at least p times), say target t, and suppose it is visited v times, where $v \geq p$. Then, the average number of visits in a revisit sequence of target t is $\frac{pn+q}{v} \leq \frac{pn+q}{p} = n + \frac{q}{p}$. We know that the maximum of a set of numbers is at least equal to the average of those numbers. Therefore, the maximum number of visits in a revisit sequence of target t is at least $n + \frac{q}{p}$. Because the number of visits is an integer, there exists revisit sequence of target t with at least $n + \lceil \frac{q}{p} \rceil$ visits; this is the desired sequence. ∎

The first two results aid in constructing walks with larger number of visits, i.e., $k \geq 2n$, using those with smaller number of visits, i.e., $n \leq k \leq 2n-1$. These two results, when combined with the results of the Diophantine Frobenius Problem (DFP) to be discussed in subsequent sections establish upper bounds on the revisit time. The third result on the other hand helps in establishing the optimality of the constructed walks by providing tight lower bounds.

A. Monotonicity of $\mathcal{R}^*{}_\Delta(k)$ for $n \leq k \leq 2n-1$

Here, we present the monotonicity property of the optimal revisit, $\mathcal{R}^*_\Delta(k)$, as a function of k, for the case $n \leq k \leq 2n-1$. *The results presented in this section hold true for any given* $\Delta \geq 0$.

Lemma 4:

a) The revisit time of a walk with at most $2n-1$ visits is equal to the travel time of the walk.
b) Optimal revisit time is a monotonic function of the number, k, of visits in the walk if $n \leq k \leq 2n-1$.

Proof:

a) For $n \leq k \leq 2n-1$, a walk with k visits contains at least one target that is visited exactly once (if not, the number of visits exceeds $2n$). The revisit sequence of such a target has the maximum travel time among all other revisit sequences in the walk. Moreover, this maximum travel time is equal to the travel time of the walk itself. Therefore, when $n \leq k \leq 2n-1$, the revisit time of a walk is equal to its travel time.
b) This result follows from triangle inequality. Consider an optimal walk $\mathcal{W}^*_\Delta(k)$ with k visits, where $n+1 \leq k \leq 2n-1$; note that $\mathcal{W}^*_\Delta(k)$ has more than one visits to at least one target. One can shortcut the visit to such a target from $\mathcal{W}^*_\Delta(k)$ to form a shortcut walk $\mathcal{SW}_\Delta(k-1)$ with $k-1$ visits. Now, from part (a) of the lemma, and the triangle inequality, it follows that $\mathcal{R}^*_\Delta(k) = \mathcal{R}(\mathcal{W}^*_\Delta(k)) = T(\mathcal{W}^*_\Delta(k)) \geq T(\mathcal{SW}_\Delta(k-1)) = \mathcal{R}(\mathcal{SW}_\Delta(k-1))$. Moreover, as $\mathcal{SW}_\Delta(k-1)$ is a feasible walk with $k-1$ visits, we have $\mathcal{R}(\mathcal{SW}_\Delta(k-1)) \geq \mathcal{R}^*_\Delta(k-1)$. Hence, $\mathcal{R}^*_\Delta(k) \geq \mathcal{R}^*_\Delta(k-1)$ for $n+1 \leq k \leq 2n-1$, and the result follows.

∎

The following corollaries to Lemma 4 are useful in the current and the following sections; proof of these corollaries are straight forward and are skipped.

1) *Corollary 1: $\mathcal{R}^*_\Delta(k) = T^*_\Delta(k)$ for $n \leq k \leq 2n-1$.*
2) *Corollary 2:*
   a) $\mathcal{W}^*_0(k)$ is optimal for any given Δ for $n \leq k \leq 2n-1$.
   b) $\mathcal{R}^*_\Delta(k) = \mathcal{R}^*_0(k) + \Delta$ for $n \leq k \leq 2n-1$.

B. Lower Bound on $\mathcal{R}^*_0(k)$, for $k \geq n$

The following is the main result of this section and is useful in establishing optimality of the solutions to be constructed in the following sections.

Lemma 5: Let $k = pn + q$, where p, q are nonnegative integers, with $p \geq 1$ and $q \leq n-1$. Then,

$$\mathcal{R}^*_0(k) \geq \mathcal{R}^*_0(n + \lceil \frac{q}{p} \rceil).$$

Proof: A lower bound on $\mathcal{R}_0^*(k)$ results from the definition of revisit sequence and Lemmas 3 and 4.

Since $k \geq n$, Euclidean division of k by n yields p as the quotient and q as the remainder, i.e., $k = pn + q$, where $p, q \in Z_+$, $p \geq 1$, and $0 \leq q \leq n-1$. It follows from Lemma 3 that $\mathcal{R}_0^*(k)$ is lower bounded by the travel time of a revisit sequence with at least $n + \lceil \frac{q}{p} \rceil$ visits. Let this sequence be denoted by RS_0. Without loss of generality (W.l.o.g), one can assume that RS_0 contains a visit to every target. If a visit to a target is missing from RS_0, one can find a larger revisit sequence that contains RS_0 and has the missing target as its terminal nodes.

Now, suppose RS_0 has exactly $n + \lceil \frac{q}{p} \rceil$ visits. Then, it follows that $\mathcal{R}_0^*(k) \geq T(\mathrm{RS}_0) \geq T_0^*(n + \lceil \frac{q}{p} \rceil) = \mathcal{R}_0^*(n + \lceil \frac{q}{p} \rceil)$. The last equality follows from the abovementioned section, as $n \leq n + \lceil \frac{q}{p} \rceil \leq 2n-1$. If RS_0 has more than $n + \lceil \frac{q}{p} \rceil$ visits, it can be shortcut to form a sequence with $n + \lceil \frac{q}{p} \rceil$ visits; due to triangle inequality, the travel time of the latter sequence is lower than that of RS_0. So, once again, it follows that $\mathcal{R}_0^*(k) \geq T(\mathrm{RS}_0) \geq T_0^*(n + \lceil \frac{q}{p} \rceil) = \mathcal{R}_0^*(n + \lceil \frac{q}{p} \rceil)$.

Therefore, given $k = pn + q$ visits, $\mathcal{R}_0^*(k)$ is lower bounded by $\mathcal{R}_0^*(n + \lceil \frac{q}{p} \rceil)$. In other words, $\mathcal{R}_0^*(k)$ is lower bounded by the revisit times of optimal walks with n to $2n-1$ visits. ■

Next, we set to only utilize optimal walks with at most $2n-1$ visits to construct optimal walks for any $k \geq 2n$. To accomplish this task, along with Lemmas 1, 2, and the lower bounds developed here, we use the solution to a special case of the DFP[4] [18]. Of particular relevance to this article is the Frobenius number of integers $n, n+1, \ldots, n+l$, a closed form of which is given by

$$F(n, n+1, n+2, \ldots, n+l) = n \left\lfloor \frac{n-2}{l} \right\rfloor + (n-1).$$

For the special case of $l = 1$, $F(n, n+1) = n(n-2) + n - 1 = n^2 - n - 1$, and hence, any integer $k \geq n^2 - n$ can be expressed as a nonnegative integer combination of just two numbers, n and $n+1$. Another special case of interest is $l = n-2$; corresponding to this value of l, we have

$$F(n, n+1, n+2, \ldots, 2n-2) = 2n-1.$$

Hence, any $k \geq 2n$ can be expressed as a nonnegative integer conic combination of $n, n+1, n+2, \ldots, 2n-2$. The following Lemma is helpful in establishing an upper bound.

Lemma 6: Suppose $k = pn + q$ with $p \geq 1$ and $0 < q \leq n-1$ being nonnegative integers. Then

$$k > F\left(n, n+1, \ldots, n + \left\lceil \frac{q}{p} \right\rceil\right).$$

Proof: Note that $p(n + \left\lceil \frac{q}{p} \right\rceil) \geq k \geq p(n + \left\lfloor \frac{q}{p} \right\rfloor)$. Arrange p jars with $n + \left\lceil \frac{q}{p} \right\rceil$ balls in each; then total number of balls is at least k. If a ball is removed from each jar, then the count of balls drops to a number not exceeding k. Hence, one can take one ball from some of the jars, if necessary, to make the total count of balls equal to k. One can then count the number of jars with $n + \left\lfloor \frac{q}{p} \right\rfloor$ and $n + \left\lceil \frac{q}{p} \right\rceil$ balls in them; say, it is α_0 of the former variety and α_1 of the latter variety. Clearly, $k = \alpha_0(n + \left\lfloor \frac{q}{p} \right\rfloor) + \alpha_1(n + \left\lceil \frac{q}{p} \right\rceil)$, where α_0, α_1 are nonnegative integers. Therefore, it is possible to arrange k as a nonnegative integer conic combination of $n + \left\lfloor \frac{q}{p} \right\rfloor, n + \left\lceil \frac{q}{p} \right\rceil$ as this is the set of count of number of balls in the jars. ■

[4]The DFP asks the following question: given relatively prime positive integers $a_1, \ldots, a_n$, find the largest positive integer that cannot be represented as a nonnegative integer linear combination of $a_1, \ldots, a_n$.

C. Proof of Theorem 1

Proof: Let $l := \left\lceil \frac{q}{p} \right\rceil$. From Lemma 6, k can be expressed as an integer conic combination of $n + l - 1$ and $n + l$; let the combination be $k = \alpha_0(n + l - 1) + \alpha_1(n + l)$, where α_0, α_1 are nonnegative integers. Consider a walk $\mathcal{W}_0^*(n+l)$ that is optimal for $n + l = n + \lceil \frac{q}{p} \rceil$ visits. If $\frac{q}{p}$ is an integer, α_0 can be set to 0. Otherwise, let $\mathcal{SW}_0(n + l - 1)$ denote a shortcut walk of $\mathcal{W}_0^*(n+l)$ with $n + l - 1$ visits. Then, a walk $\mathcal{W}_0(k)$ with k visits can be constructed by concatenating in the following order α_1 walks of $\mathcal{W}_0^*(n+l)$ followed by α_0 walks of $\mathcal{SW}_0(n + l - 1)$. By Lemmas 1 and 2, it is clear that $\mathcal{R}(\mathcal{W}_0(k)) = \mathcal{R}({\mathcal{W}_0}^*(n+l))$. Since $\mathcal{W}_0(k)$ is a feasible walk of k visits, its revisit time is no less than that of the optimum, and hence, $\mathcal{R}(\mathcal{W}_0^*(n+l)) = \mathcal{R}(\mathcal{W}_0(k)) \geq \mathcal{R}(\mathcal{W}_0^*(k))$. From Lemma 5, we have $\mathcal{R}_0^*(k) \geq \mathcal{R}_0^*(n+l)$. Therefore, $\mathcal{R}_0^*(k) = \mathcal{R}_0^*(n+l) = \mathcal{R}_0^*(n + \lceil \frac{q}{p} \rceil)$. ■

Remarks:

1) Proof for the abovementioned theorem embeds a procedure to construct optimal walks with higher number of visits using those with lower number of visits via concatenation; it can be seen from the proof that in order *to determine an optimal walk with k visits, it is sufficient to explicitly compute an optimal walk with* $n + \lceil \frac{q}{p} \rceil$ *visits.* The construction procedure will be used in Section VI to determine optimal walks with large number of visits.
2) The integer conic combination of $n, n+1, \ldots, n + \lceil \frac{q}{p} \rceil$ to obtain k is not unique; correspondingly, there are several walks with the same revisit time but can have different travel times. There are at least two consequences, as follows.
   a) The idea of a s-Frobenius number [19] of integers $a_1, \ldots, a_n$ generalizes the notion of a Frobenius number—it is the largest integer that cannot be expressed as an integer conic combination of $a_1, a_2, \ldots, a_n$ in at least s different ways. The 2-Frobenius number, F_2, of $n, n+1, \ldots, n+l$ is given by

$$F_2(n, n+1, \ldots, n+l) = n \left\lfloor \frac{n}{l} \right\rfloor + (n+1).$$

   In particular for $l = 1$, it follows that every $k \geq n^2 + n + 1$ can be expressed as an integer conic combination in at least two different ways.

   In surveillance applications, this possibility can be exploited by randomly concatenating an optimal walk of $n + l$ visits with its shortcut walks without increasing the revisit time, thereby reducing the predictability of

UAVs paths for monitoring. An example is provided in the numerical results section.

b) This can compound the difficulty of planning the UAV motion when fuel limitation is specified in terms of the maximum distance that a UAV can travel before refueling/recharging. The problem of finding a minimum distance walk for a specified walk revisit time is a difficult problem and is NP-hard (as it is a generalization of TSP).

Simple corollaries of the abovementioned theorem with consequences for computing optimal walks are given as follows.

Corollary 3: For $k \geq 2n$, $\mathcal{R}_0^*(k) \subset \{\mathcal{R}_0^*(n), \mathcal{R}_0^*(n+1), \ldots, \mathcal{R}_0^*(2n-1)\}$.

Proof: Let $k = np + q$ as before; clearly, $\left\lceil \frac{q}{p} \right\rceil \in \{0, 1, 2, \ldots, n-1\}$ as $0 \leq q \leq n-1$ and $p \geq 1$. From Theorem 1, it readily follows that $\mathcal{R}_0^*(k) = \mathcal{R}_0^*(n + \left\lceil \frac{q}{p} \right\rceil) \subset \{\mathcal{R}_0^*(n), \mathcal{R}_0^*(n+1), \ldots, \mathcal{R}_0^*(2n-1)\}$. ■

Corollary 4: For $k \geq n^2 - n$,

$$\mathcal{R}_0^*(k) = \begin{cases} \mathcal{R}_0^*(n), & \text{when } k \text{ is an integer multiple of } n, \\ \mathcal{R}_0^*(n+1), & \text{otherwise.} \end{cases}$$

Proof: For $k \geq n^2 - n$, when k is expressed as $pn + q$ with $p, q \in Z_+$, $p \geq n-1$, and $q \leq n-1$, observe that $q \leq p$. In this case, $n + \lceil \frac{q}{p} \rceil$ takes only two values: n when k is an integer multiple of n; and $n+1$ otherwise. Therefore, from Theorem 1, the corollary follows. ■

Remarks:

1) Recall from the proof of Theorem 1 that it is sufficient to explicitly compute $\mathcal{W}_0^*(n + \lceil \frac{q}{p} \rceil)$ to determine an optimal walk with k visits. Now, as shown in the proof of corollary 3, when $k \geq 2n$, we have $n + \lceil \frac{q}{p} \rceil \in \{n, \ldots, 2n-1\}$. Therefore, it follows that *an optimal walk with k visits, where $k \geq 2n$, can be determined by explicitly computing one among the following set: $\{\mathcal{W}_0^*(n), \ldots, \mathcal{W}_0^*(2n-1)\}$.*
2) Similarly, as shown in the proof of corollary 4, when $k \geq n^2 - n, n + \lceil \frac{q}{p} \rceil \in \{n, n+1\}$. Therefore, *when $k \geq n^2 - n$, an optimal walk with k visits can be determined by explicitly computing either $\mathcal{W}_0^*(n)$ or $\mathcal{W}_0^*(n+1)$.*
3) Furthermore, when k is an integer multiple of n, we have $n + \lceil \frac{q}{p} \rceil = n$. Therefore, *an optimal walk with k visits, when k is an integer multiple of n, can be determined by computing $\mathcal{W}_0^*(n)$ alone, which is an optimal TSP tour over the targets.*

The abovementioned results lead to a significant savings in the computational effort required to solve the problem for the following reasons: first, it is relatively easy to compute optimal walks with a smaller number of visits; second, when the number visits is at most $2n-1$, the revisit time of a walk is equal to its travel time, and the problem of finding a walk with the least revisit time reduces to that of finding a walk with the least travel time. The latter is a well-studied problem, and standard techniques can be employed to solve the same. In Appendix B, we present an efficient formulation for the swift computation of optimal walks with at most $2n-1$ visits. These walks can later be concatenated to construct optimal walks for any given $k \geq 2n$ using the construction procedure provided in the Proof of Theorem 1.

D. Concatenating a Different Walk After Servicing

Due to the reasons discussed earlier, we choose to repeat the same k-visit walk after servicing the UAV. However, concatenating different k-visit walks after servicing the UAV can sometimes provide a lower revisit time (under the assumption that these different walks, after concatenation, are repeated). Here, we present cases in which repeating an optimal k-visit walk after every service provides the lowest revisit time and one cannot obtain a lower revisit time by concatenating different k-visit walks. We begin with the case $k \leq \frac{3n-1}{2}$.

Lemma 7: Let $\mathcal{W}_0^1(k)$ and $\mathcal{W}_0^2(k)$ be two different walks with k visits each, where $k \leq \frac{3n-1}{2}$. Then, $\mathcal{R}(\mathcal{W}_0^1(k) \circ \mathcal{W}_0^2(k)) \geq \mathcal{R}_0^*(k)$.

Proof: When $k \leq \frac{3n-1}{2}$, total number of visits in the concatenated walk, $\mathcal{W}_0^1(k) \circ \mathcal{W}_0^2(k)$, is at most $3n-1$ visits. Then, there is at least one target, say target t, that is visited at most twice in $\mathcal{W}_0^1(k) \circ \mathcal{W}_0^2(k)$. Because both $\mathcal{W}_0^1(k)$ and $\mathcal{W}_0^2(k)$ are feasible walks, they have exactly one visit each to target t. Therefore, $\mathcal{W}_0^1(k) \circ \mathcal{W}_0^2(k)$ has only two revisit sequences corresponding to target t; let the travel time of these sequences be t_1 and t_2. Clearly, $t_1 + t_2 = T(\mathcal{W}_0^1) + T(\mathcal{W}_0^2)$. Then, from Lemma 4, it follows that $t_1 + t_2 = \mathcal{R}(\mathcal{W}_0^1) + \mathcal{R}(\mathcal{W}_0^2)$, which implies $\max\{t_1, t_2\} \geq \min\{\mathcal{R}(\mathcal{W}_0^1), \mathcal{R}(\mathcal{W}_0^2)\} \geq \mathcal{R}_0^*(k)$. Furthermore, from the definition of revisit time, it follows that $\mathcal{R}(\mathcal{W}_0^1(k) \circ \mathcal{W}_0^2(k)) \geq \max\{t_1, t_2\} \geq \mathcal{R}_0^*(k)$. Hence, $\mathcal{R}(\mathcal{W}_0^1(k) \circ \mathcal{W}_0^2(k)) \geq \mathcal{R}_0^*(k)$. ■

Before discussing other cases, we first present bounds on the revisit time of a walk, $\mathcal{NW}_0(mk)$, obtained by concatenating arbitrary k-visit walks after each servicing cycle (supposing that there are m servicing cycles in the mission). Note that $\mathcal{NW}_0(mk)$ is a feasible PMP walk with mk visits. Therefore, the revisit time of $\mathcal{NW}_0(mk)$ is at least $\mathcal{R}_0^*(mk)$. Also note that a walk obtained by repeating an optimal PMP walk with k visits for m times is a feasible walk for the case of concatenating arbitrary walks after every servicing cycle; from Lemma 1, the revisit time of the former walk is $\mathcal{R}_0^*(k)$. Therefore, the revisit time of $\mathcal{NW}_0(mk)$ is upper bounded by $\mathcal{R}^*(k)$ and lower bounded by $\mathcal{R}_0^*(mk)$.

Now, consider the case when k is large, say $k \geq n^2 - n$. This case can be divided into the following three subcases.

1) k is an integer multiple of n (and therefore, mk is an integer multiple of n).
2) mk is not an integer multiple of n (and therefore k is not an integer multiple of n).
3) k is not an integer multiple of n and mk is an integer multiple of n.

Recall from the remarks following Corollary 4 that when k is an integer multiple of n, $\mathcal{R}_0^*(k) = \mathcal{R}_0^*(n)$, and otherwise, $\mathcal{R}_0^*(k) = \mathcal{R}_0^*(n+1)$. Therefore, in subcase (1), we have $\mathcal{R}_0^*(k) = \mathcal{R}_0^*(mk) = \mathcal{R}_0^*(n)$, and in subcase (2), we have we have $\mathcal{R}_0^*(k) = \mathcal{R}_0^*(mk) = \mathcal{R}_0^*(n+1)$. Since the upper and lower bounds of the revisit time of $\mathcal{NW}_0(mk)$ match in these cases, and are equal to $\mathcal{R}_0^*(k)$, we can conclude that it is *optimal to repeat a PMP walk with k visits* after every servicing. Subcase (3) is inconclusive, however, as the upper and lower bounds for this case are $\mathcal{R}_0^*(n+1)$ and $\mathcal{R}_0^*(n)$, respectively, and $\mathcal{R}_0^*(n+1) \geq$

$\mathcal{R}_0^*(n)$. Similar bounds can be constructed for other cases, i.e., for $k < n^2 - n$.

V. Solution to PMP for Nonnegligible Service Times

In this section, we first consider the case with service time $\Delta \geq 2c_{\min}$, where $c_{\min} = \min_{i,j \in \mathcal{T}} \; c(i,j)$. For this case, we show that the optimal revisit time is asymptotically unimodal; specifically, we show that $\mathcal{R}_\Delta^*(k) = \mathcal{R}_\Delta^*(n)$ for $k \geq n^2 + n$.

Similar to the previous section, we first show that $\mathcal{R}_\Delta^*(k)$ is lower bounded by $\mathcal{R}_\Delta^*(n)$. Then, we provide a procedure to construct feasible walks with k visits, for $k \geq n^2 + n$, such that their revisit times match the lower bounds.

A. Lower Bound on Service Time

Lemma 8: Given $k \geq n$ and $\Delta \geq 0$, $\mathcal{R}_\Delta^*(k) \geq \mathcal{R}_\Delta^*(n)$.

Proof: Suppose $\mathcal{W}_\Delta^*(k)$ is an optimal walk with k visits when there is a service time of Δ at the depot. Let RS_Δ be a revisit sequence of $\mathcal{W}_\Delta^*(k)$ that has the servicing visit to the depot. W.l.o.g, one can assume that RS_Δ has visits to all the targets; suppose the visit to a target is missing from the chosen sequence, one can reselect a larger revisit sequence that contains the earlier one and has the missing target as its terminal nodes. From the definition of revisit sequence, it follows that $\mathcal{R}_\Delta^*(k) \geq T(\mathrm{RS}_\Delta)$. Since RS_Δ has visits to all the targets, its repeated visits to targets, if any, can be shortcut to form a walk with n visits. So, by triangle inequality, it follows that $T(\mathrm{RS}_\Delta) \geq T_\Delta^*(n)$. Besides, from Corollary 1, we have that $T_\Delta^*(n) = \mathcal{R}_\Delta^*(n)$. Therefore, combining the abovementioned equalities and inequalities, it follows that $\mathcal{R}_\Delta^*(k) \geq \mathcal{R}_\Delta{}^*(n)$. ■

B. Construction Procedure

Next, given $k \geq n^2 + n$, we show that an optimal walk with k visits can be constructed by explicitly computing only $\mathcal{W}_0^*(n)$, which is an optimal TSP tour. Utilizing $\mathcal{W}_0^*(n)$, we first generate intermediate walks $\mathcal{W}_\Delta(n)$ and $\mathcal{AW}_0(n+1)$ with n and $n+1$ visits, respectively, that aid in the construction process.

$\mathcal{W}_\Delta(n)$ has a sequence of visits that is identical to $\mathcal{W}_0^*(n)$; however, the visit to the depot in $\mathcal{W}_\Delta(n)$ has a servicing time of Δ. Note from Corollary 2 that $\mathcal{W}_\Delta(n)$ is an optimal walk with n visits when the servicing time is Δ. So, we have $\mathcal{R}(\mathcal{W}_\Delta(n)) = \mathcal{R}_\Delta^*(n)$.

$\mathcal{AW}_0(n+1)$ is obtained by augmenting visits to $\mathcal{W}_0^*(n)$ according to the procedure discussed as follows; the same is illustrated in Fig. 7 for a sample instance with five targets and an optimal TSP tour given by $\mathcal{W}_0^*(n) = \{d, 1, 4, 3, 2, d\}$.

1) Identify the edge $e = \{i, j\}$ with the least cost (i.e., $c(i,j) = c_{\min}$); in the illustrative example, this corresponds to the edge $\{3, 4\}$.
2) Double the least cost edge and augment it to $\mathcal{W}_0^*(n)$ (see the black edge $\{3, 4\}$ and the red dotted edge $\{3, 4\}$). The augmented walk now has $n + 2$ visits.
3) Shortcut the augmented walk to obtain $\mathcal{A}W_0$ with $n + 1$ visits (as in the red dotted edges $\{4, 3\}, \{3, 2\}$ being shortcut by the solid black edge $\{4, 2\}$ in Fig. 7).

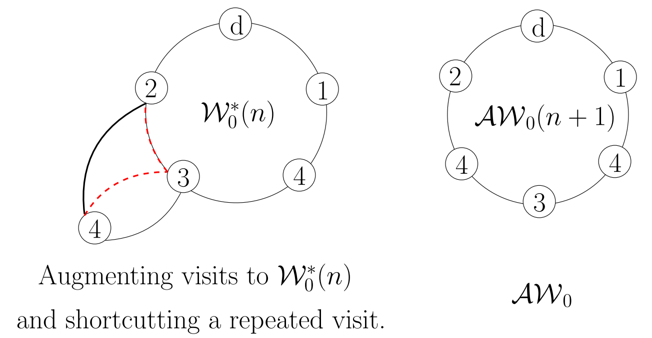


Fig. 7. Construction of $\mathcal{AW}_0$, a walk that is required for the construction of optimal walks with $k \geq n^2 + n$ visits for the case $\Delta \geq 2c_{\min}$. Here, $\mathcal{AW}_0$ is constructed by augmenting the least cost edges (between targets 3 and 4) to $\mathcal{W}_0^*(n)$ and shortcutting a repeated visit from the augmented walk.

In the illustrative figure, the resultant walk $\mathcal{AW}_0(6) = \{d, 1, 4, 3, 4, 2, d\}$ is shown on the right.

The revisit time of $\mathcal{A}W_0$ can be upper bounded with the help of the following lemma.

Lemma 9: $\mathcal{R}(\mathcal{AW}_0) \leq \mathcal{R}_\Delta^*(n)$

Proof: It follows from Lemma 4, its corollaries and the abovementioned construction process that

$$\begin{aligned}\mathcal{R}(\mathcal{AW}_0) = T(\mathcal{AW}_0) &\leq T(\mathcal{W}_0^*(n)) + 2c_{\min}\\ &\leq T_0^*(n) + \Delta = T_\Delta^*(n) = \mathcal{R}_\Delta^*(n).\end{aligned}$$

■

Now we have three elementary walks $\mathcal{AW}_0(n+1)$, $\mathcal{W}_0^*(n)$, and $\mathcal{W}_\Delta(n)$, with the sequence of visits in the same order except for the repeated visit in $\mathcal{AW}_0(n+1)$. The maximum of revisit times of these walks is $\mathcal{R}_\Delta^*(n)$. With the help of these walks, we prove the main result of this section, which is given by the following theorem. For a simpler notation, here we refer to $\mathcal{AW}_0(n+1)$ as $\mathcal{AW}_0$, $\mathcal{W}_0^*(n)$ as $\mathcal{W}_0^*$, and $\mathcal{W}_\Delta(n)$ as $\mathcal{W}_\Delta$.

Theorem 2: Given $k \geq n^2 + n$ and $\Delta \geq 2c_{\min}$, $\mathcal{R}_\Delta^*(k) = \mathcal{R}_\Delta^*(n)$.

Proof: Recall from the DFP that any integer $k_1 \geq n^2 - n$ can be written as an integer conic combination of n and $n+1$; say $k_1 = \beta_0 n + \beta_1(n+1)$, where $\beta_0, \beta_1 \in Z_+$. Consequently, given any number of visits, k, such that $k \geq n^2 + n$, it can be expressed as $k = n + k_1 + n = n + \beta_0 n + \beta_1(n+1) + n$.

Then, a walk $\mathcal{CW}_\Delta(k)$ with k visits can be constructed by concatenating in the following order one walk of $\mathcal{W}_\Delta$ (which includes the servicing visit to the depot), β_0 walks of $\mathcal{W}_0^*$, β_1 walks of $\mathcal{AW}_0$, followed by one walk of $\mathcal{W}_0^*$, i.e.,

$$\mathcal{CW}_\Delta(k) = \mathcal{W}_\Delta \circ (\underbrace{\mathcal{W}_0^* \circ \cdots \circ \mathcal{W}_0^*}_{\beta_0 \text{ times}} \circ \underbrace{\mathcal{AW}_0 \circ \cdots \circ \mathcal{AW}_0}_{\beta_1 \text{ times}}) \circ \mathcal{W}_0^*.$$

Alternatively, $\mathcal{CW}_\Delta(k)$ can also be expressed as $\mathcal{W}_\Delta \circ \mathcal{IW}_0(k-n)$, where $\mathcal{IW}_0(k-n) = (\underbrace{\mathcal{W}_0^* \circ \cdots \circ \mathcal{W}_0^*}_{\beta_0 \text{ times}} \circ \underbrace{\mathcal{AW}_0 \circ \cdots \circ \mathcal{AW}_0}_{\beta_1 \text{ times}}) \circ \mathcal{W}_0^*$. Note that $\mathcal{IW}_0(k-n)$ is formed by concatenating only the walks $\mathcal{AW}_0$ and $\mathcal{W}_0^*$; both $\mathcal{AW}_0$ and $\mathcal{W}_0^*$ have zero service time, and the latter can be obtained by shortcutting the repeated visit from the former. So, from Lemmas 1 and 2, it follows that

$\mathcal{R}(\mathcal{IW}_0(k-n)) = \mathcal{R}(\mathcal{AW}_0) \leq \mathcal{R}^*_\Delta(n)$; the last inequality follows from Lemma 9.

Then, the only additional revisit sequences in $\mathcal{CW}_\Delta(k)$ compared to those in $\mathcal{IW}_0$ are those obtained by concatenating $\mathcal{W}_0^*$ and $\mathcal{W}_\Delta$. Because $\mathcal{W}_0^*$ and $\mathcal{W}_\Delta$ have identical sequences of visits, and the only difference is the servicing time Δ, it follows that the travel times of the new revisit sequences are at most $\mathcal{R}^*_\Delta(n)$. Therefore, it follows that $\mathcal{R}(\mathcal{CW}_\Delta(k)) \leq \mathcal{R}^*_\Delta(n)$.

As $\mathcal{CW}_\Delta(k)$ is a feasible walk of k visits and its revisit time is no less than that of the optimum, we have $\mathcal{R}^*_\Delta(k) \leq \mathcal{R}(\mathcal{CW}_\Delta(k)) \leq \mathcal{R}^*_\Delta(n)$. Besides, from Lemma 8, we have $\mathcal{R}^*_\Delta(k) \geq \mathcal{R}^*_\Delta(n)$. Therefore, $\mathcal{R}^*_\Delta(k) = \mathcal{R}^*_\Delta(n)$. ■

Results of this section can be summarized as follows: For $k \geq n^2 + n$,

1) $\mathcal{R}^*_\Delta(k) = \mathcal{R}^*_\Delta(n)$;
2) determining an optimal walk with k $(\geq n^2 + n)$ visits requires the explicit computation of only $\mathcal{W}_0^*(n)$, which is an optimal TSP tour over all the targets (including the depot);
3) an optimal walk with k visits is given by $\mathcal{CW}_\Delta(k)$.

C. Case When the Servicing Time is Any Positive Real Number, i.e., $\Delta > 0$

For any given k, optimal solutions from Section IV (i.e., $\mathcal{W}_0^*(k)$) form feasible solutions for this case, and their revisit times are at most Δ units away from the optimum. This can be seen from the following inequalities:

$$\begin{aligned}\mathcal{R}^*_0(k) = \mathcal{R}(\mathcal{W}_0^*(k)) \leq \mathcal{R}^*_\Delta(k) \leq \mathcal{R}(\mathcal{W}_0^*(k))\\ + \Delta = \mathcal{R}^*_0(k) + \Delta.\end{aligned}$$

Note that this result holds true for any positive Δ.

VI. Numerical Simulations

In this section, we present the results of numerical simulations performed on 800 instances with n ranging from 5 to 50. These simulations are used to accomplish the following.

1) Corroborate the theoretical results concerning the structure of optimal solutions developed in this article.
2) Illustrate the construction procedure discussed in Sections IV and V through an example.
3) Show the improvement in the computation time due to the results developed in this article.

The target locations in each instance were generated independently and uniformly from a grid sized 20×20 using the rand function in Julia [20]. The UAV was assumed to travel at a uniform speed of 1 unit per unit time. The travel time between pairs of targets were set equal to the Euclidean distance between them.

First, to corroborate the results developed in this article, we made use of the MBLP formulation presented in [21] to compute optimal PMP solutions. For every generated instance, the formulation was solved in Gurobi [22], a commercially available solver, from n visits to a number of visits for which the formulation was solvable in a reasonable time. Additionally, for every instance, the ILP formulation presented in the Appendix was used to find walks with the minimum travel time for n to $2n-1$ visits. All the simulations were implemented in Julia, using the package JuMP [23], on a MacBook Pro with 16 GB RAM, Inter Core i7 processor and a processor speed of 2.5 GHz.

As discussed in Section I, the computation time required to solve the MBLP formulation increases with the increase in the number of targets and the number of visits. Here, it was difficult to solve the MBLP formulation beyond $n^2 - n$ visits for instances with more than 6 targets. In order to corroborate the results proved in earlier sections, we consider instances with upto six targets. For each instance, optimal solutions and the optimal revisit times corresponding to different numbers of visits were noted for the cases $\Delta = 0$ and $\Delta \geq 2c_{\min}$. For every instance, the optimal revisit times are plotted against the number of visits in the walk; Sample plots for instances with 4, 5, and 6 targets are presented in Figs. 8 –10, respectively for the case $\Delta = 0$ and Figs. 11 –13, respectively, for the case $\Delta = 2c_{\min}$.

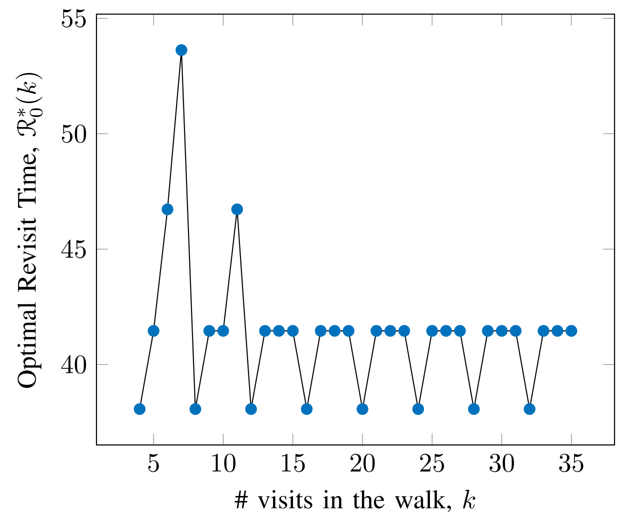


Fig. 8. Structure of optimal revisit time for an instance with four targets when $\Delta = 0$.

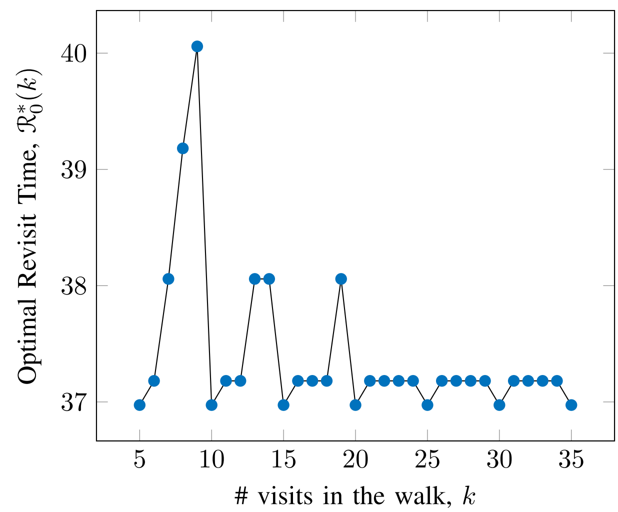


Fig. 9. Structure of optimal revisit time for an instance with five targets when $\Delta = 0$.

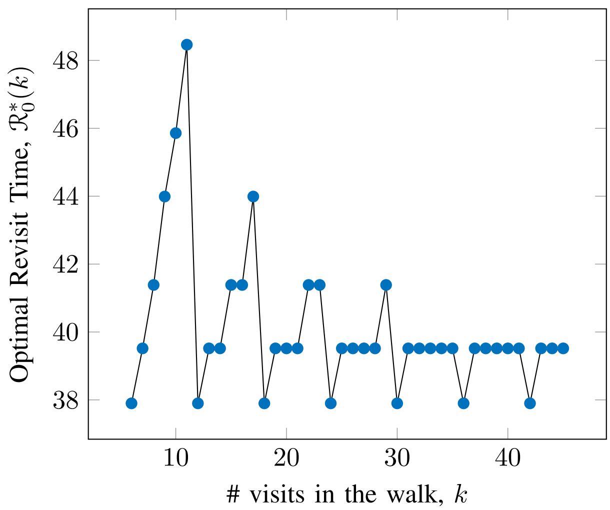


Fig. 10. Structure of optimal revisit time for an instance with six targets when $\Delta = 0$.

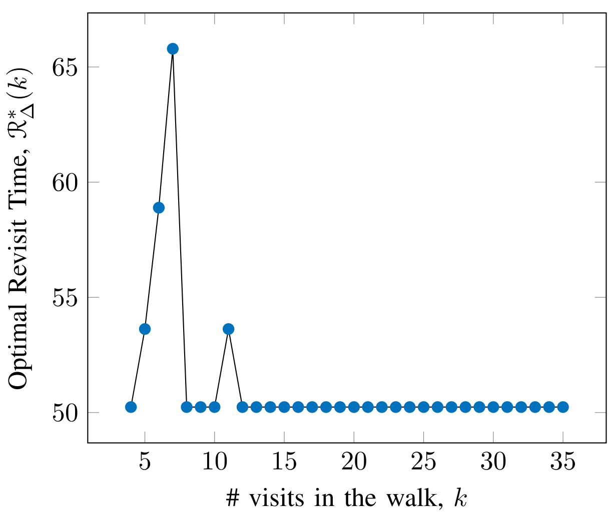


Fig. 11. Structure of optimal revisit time for an instance with four targets and $\Delta = 2 \times c_{\min}$, where $c_{\min} = 6.08$ units.

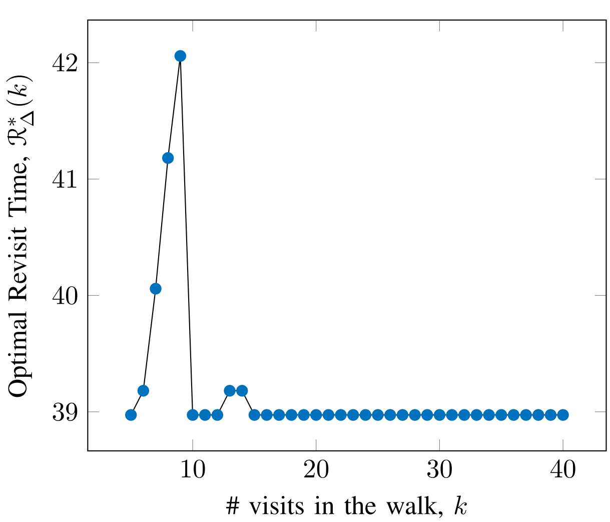


Fig. 12. Structure of optimal revisit time for an instance with five targets and $\Delta = 2 \times c_{\min}$, where $c_{\min} = 1$ unit.

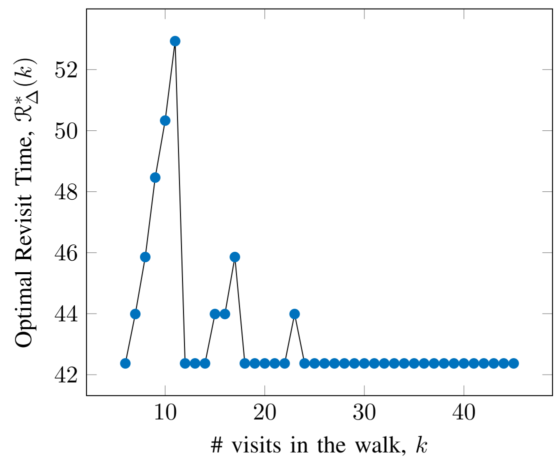


Fig. 13. Structure of optimal revisit time for an instance with six targets and $\Delta = 2 \times c_{\min}$, where $c_{\min} = 2.24$ units.

A. *Plots of Optimal Revisit Time*

The following observations were made from the plots.

Case 1: $\Delta = 0$

1) $\mathcal{R}_0^*(k) \geq \mathcal{R}_0^*(k+n), \quad \forall k \geq n$.
2) $\mathcal{R}_0^*(k)$ is lower bounded by $\mathcal{R}_0^*(n)$ for any given $k \geq n$.
3) $\mathcal{R}_0^*(n)$ is equal to the optimal cost of an optimal TSP tour over the targets.
4) $\mathcal{R}_0^*(k)$ takes at most n distinct values: $\mathcal{R}_0^*(n)$, $\mathcal{R}_0^*(n+1), \ldots, \mathcal{R}_0^*(2n-1)$; specifically, $\mathcal{R}_0^*(k) = \mathcal{R}_0^*(n + \lceil \frac{q}{p} \rceil)$ when k is expressed as $pn + q$ with $p, q \in Z_+$, $p \geq 1$, $0 \leq q \leq n-1$.
5) For $n \leq k \leq 2n-1$, $\mathcal{R}^*(k)$ is a monotonic function of k and $\mathcal{R}_0^*(k) = T_0^*(k)$; the latter result was observed by comparing optimal values for the MBLP and the ILP formulations.

The structure was *independent of the relative locations of the targets*, but was only dependent on the number of targets in an instance. Furthermore, as the number of allowed visits increased, visits to targets in optimal walks were observed to be uniformly spread out as one would expect for a min–max objective.

Case 2: $\Delta \geq 2c_{\min}$

1) $\mathcal{R}_\Delta^*(k)$ is lower bounded by $\mathcal{R}_\Delta^*(n)$ for all $k \geq n$.
2) $\mathcal{R}_\Delta^*(k) = \mathcal{R}_0^*(k) + \Delta$ for $n \leq k \leq 2n-1$.
3) $\mathcal{R}_\Delta^*(k) = \mathcal{R}_\Delta^*(n)$ for $k \geq n^2 + n$ visits.

In simulations, the asymptotic unimodality was observed to commence much earlier than the proposed $n^2 + n$ visits. A reason for this is one being able to construct elementary walks similar to $\mathcal{AW}$ with number of visits greater than $n+1$, i.e., $n+2$ visits and higher, but a revisit time of at most $\mathcal{R}_\Delta^*(n)$. Then, from the Frobenius number of $n, n+1, n+2$, and so on, one can concatenate these elementary walks to construct optimal walks with a revisit time of $\mathcal{R}_\Delta^*(n)$ for k smaller than $n^2 + n$ visits.

B. *Representative Example With Five Targets*

For a representative five target scenario shown in Fig. 14, we illustrate the construction of optimal walks with $k \geq 2n$ visits and the estimation of $\bar{k}$ for a prescribed maximal travel

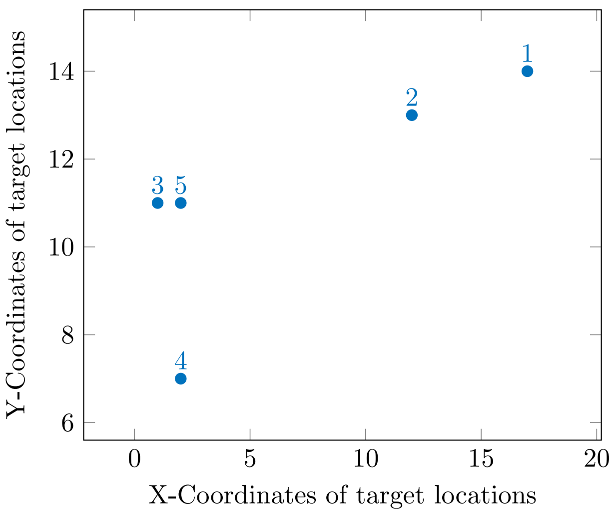


Fig. 14. Coordinates of target locations for a sample five-target instance. Note that the targets in this figure are notequidistant. However, the procedure proposed to estimate $\bar{k}$ can still be applied to such instances as discussed in Section VI-B3.

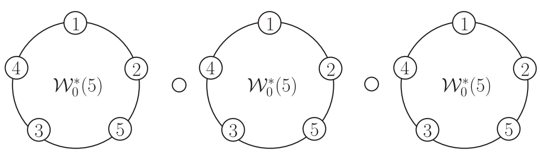


Fig. 15. Construction of optimal walks when k is an integer multiple of n; when $n = 5$ and $k = 15$, an optimal walk is given by $\mathcal{W}_0^*(15) = \mathcal{W}_0^*(5) \circ \mathcal{W}_0^*(5) \circ \mathcal{W}_0^*(5)$.

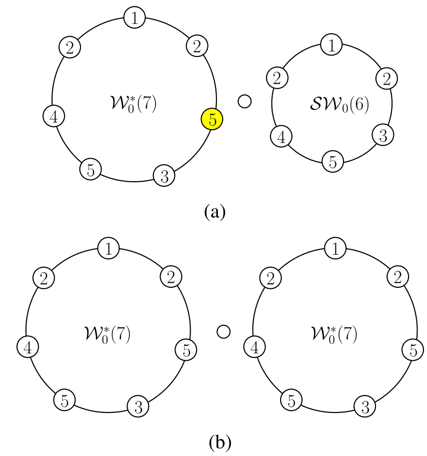


Fig. 16. Construction of optimal walks for the case $2n \leq k \leq n^2 - n$.

time (fuel capacity) of the UAV. The construction requires the knowledge of optimal walks with five–seven visits; here, these walks were computed using the ILP formulation to be the following: $\mathcal{W}_0^*(5) = (1,2,5,3,4,1)$, $\mathcal{W}_0^*(6) = (1,2,5,3,4,2,1)$, and $\mathcal{W}_0^*(7) = (1,2,5,3,5,4,2,1)$. Besides, the MBLP formulation was used to compute optimal revisit times for $5 \leq k \leq 35$ visits to verify the optimality of the constructed walks. The plot of the optimal revisit time against the number of visits in the walk for the case $\Delta = 0$ is shown in Fig. 9 and for the case $\Delta = 2c_{\min}$ is shown in Fig. 12.

1) Construction of Optimal Walks With $k \geq 2n$ Visits for $\Delta = 0$: Given any k such that $k \geq 2n$, it can be expressed as $pn + q$, where $p, q \in Z_+$, $p \geq 1$, and $0 \leq q \leq 1$. Then, from Theorem 1 it follows that $\mathcal{R}_0^*(k) = \mathcal{R}_0^*(n + \lceil \frac{q}{p} \rceil)$. Furthermore, from the proof of Lemma 6, k can be expressed as an integer conic combination of $n + \lceil \frac{q}{p} \rceil$ and $n + \lfloor \frac{q}{p} \rfloor$. While there may be multiple such combinations, here we consider a combination that is obtained by expressing q as $sp + r$ for positive integers s and r such that $0 \leq r \leq p - 1$. Then, k can be expressed as an integer conic combination of $n + \lceil \frac{q}{p} \rceil$ and $n + \lfloor \frac{q}{p} \rfloor$ as the following: $k = r(n + \lceil \frac{q}{p} \rceil) + (p - r)(n + \lfloor \frac{q}{p} \rfloor)$.

First, when k is an integer multiple of n, i.e., for $k = pn$ and $q = 0$, an optimal walk can simply be constructed by concatenating p copies of $\mathcal{W}_0^*(5)$. For example, for $k = 15$, an optimal walk can be obtained as $\mathcal{W}_0^*(15) = \mathcal{W}_0^*(5) \circ \mathcal{W}_0^*(5) \circ \mathcal{W}_0^*(5)$, as shown in Fig. 15.

Next, consider any k in the range $2n \leq k \leq n^2 - n$ and express it as $pn + q$. Then, further express q as $ps + r$, where $p, q, r \in Z_+$, $0 \leq q \leq n - 1$, $0 \leq r \leq p - 1$. If $r = 0$, i.e., if q is an integer multiple of p, then we have $n + \lceil \frac{q}{p} \rceil = n + \lfloor \frac{q}{p} \rfloor$. Then, k can simply be expressed as a multiple of $n + \lceil \frac{q}{p} \rceil$, i.e., $k = p(n + \lceil \frac{q}{p} \rceil)$. Then, an optimal walk with k visits can be obtained by concatenating p copies of $\mathcal{W}_0^*(n + \lceil \frac{q}{p} \rceil)$. For example, if $k = 14$, it can be expressed as $7 + 7$, i.e., we have $p = 2$, $q = 4$, and $n + \lceil \frac{q}{p} \rceil = 7$. So, an optimal walk with 14 visits can be obtained as $\mathcal{W}_0^*(14) = \mathcal{W}_0^*(7) \circ \mathcal{W}_0^*(7)$, as shown in Fig. 16. If q is not an integer multiple of p, then the expression for k remains $k = r(n + \lceil \frac{q}{p} \rceil) + (p - r)(n + \lfloor \frac{q}{p} \rfloor)$. Then, an optimal walk with k visits is obtained by concatenating r copies of an optimal walk with $n + \lceil \frac{q}{p} \rceil$ visits followed by $p - r$ copies of its shortcut walk with $n + \lfloor \frac{q}{p} \rfloor$ visits. For example, if $k = 13$, it can be expressed as $7 + 6$ we have $p = 2$, $q = 3$, $n + \lceil \frac{q}{p} \rceil = 7$, $n + \lfloor \frac{q}{p} \rfloor = 6$ and $r = 1$. So, an optimal walk with 13 visits, as shown in Fig. 16(a), is given by $\mathcal{W}_0^*(13) = \mathcal{W}_0^*(7) \circ \mathcal{SW}_0(6)$, where $\mathcal{SW}_0(6) = (1,2,3,5,4,2,1)$ is obtained by shortcutting the repeated visit to target 5 from $\mathcal{W}_0^*(7)$.

When $k \geq n^2 - n$, from Corollary 4, the construction procedure does not require the knowledge of optimal walks with more than six visits. The case with k being an integer multiple of n has been discussed earlier. When k is not an integer multiple of n and $k \geq n^2 - n$, k can always be expressed as $q(n+1) + (p - q)(n)$. So, an optimal walk with k visits can always be constructed by concatenating q copies of an optimal walk with $n + 1$ visits followed by $p - q$ copies of its shortcut walk with n visits. For example, $k = 21$ and $k = 23$ can be expressed as $6 + 3(5)$ and $3(6) + 5$, respectively. So, as shown in Fig. 17, optimal walks with 21 and 23 visits can be obtained as $\mathcal{W}_0^*(6) \circ \mathcal{SW}_0(5) \circ \mathcal{SW}_0(5) \circ \mathcal{SW}_0(5)$ and $\mathcal{W}_0^*(6) \circ \mathcal{W}_0^*(6) \circ \mathcal{W}_0^*(6) \circ \mathcal{SW}_0(5)$, respectively, where $\mathcal{SW}_0(5)$ is obtained by shortcutting the repeated visit from $\mathcal{W}^*(6)$.

2) Construction of Optimal Walks With $k \geq n^2 + n$ Visits When $\Delta \geq 2c_{\min}$: When $\Delta = 2c_{\min}$, we use $\mathcal{W}_0^*(5)$ to construct optimal walks with $k \geq n^2 + n = 30$ visits. This construction requires the generation of an elementary walk $\mathcal{AW}_0$

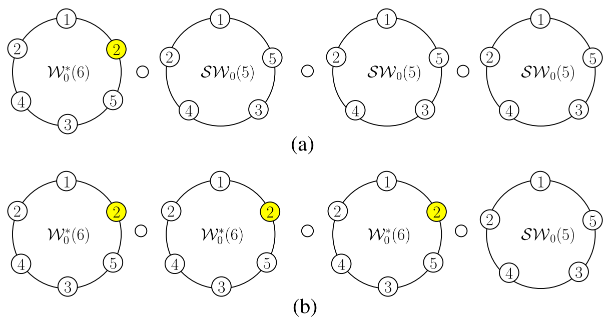


Fig. 17. Construction of optimal walks for the case $k \geq n^2 - n$.

with six visits. For the considered instance, the least cost edge is given by (3,5). So, following the procedure in Section V, $\mathcal{AW}_0$ can be constructed by appending visits (3,5) and (5,3) to $\mathcal{W}_0^*(5)$ and shortcutting a repeated visit; here, the resultant walk is obtained as $\mathcal{AW}_0(6) = (1, 2, 3, 5, 3, 4, 1)$ by shortcutting a revisit to target 5 . Now, given any $k \geq n^2 + n$, one can express $k - 2n$ as $pn + q$, where p, q are positive integers with $0 \leq q \leq n - 1$. Then, k can be expressed as $n + q(n + 1) + (p - q)(n) + n$. For example, $k = 33$ can be expressed as $k = 5 + 5 + 3(6) + 5$. So, from the proof of Theorem 2, an optimal walk with 33 visits is given by $\mathcal{W}_\Delta(5) \circ \mathcal{W}_0^*(5) \circ \mathcal{AW}_0(6) \circ \mathcal{AW}_0(6) \circ \mathcal{AW}_0(6) \circ \mathcal{W}_0^*(5)$, where $\mathcal{W}_\Delta(5)$ has the same sequence of visits as of $\mathcal{W}_0^*(5)$, but its first element is associated with a service time of Δ.

The revisit times of all the solutions constructed earlier match the optimal revisit times evaluated using the MBLP formulation, corroborating the results developed in this article. For larger instances, as it is difficult to solve the MBLP, one can find optimal walks with n to $2n - 1$ visits using the ILP formulation, and use these solutions to construct optimal walks with greater number of visits. Unlike, the MBLP, the ILP is has low computation time, which does not necessarily increase with the number of visits in the walk. Its superiority over the MBLP formulation in computing optimal PMP walks will be demonstrated in the following section through the average computation times for large instances. Before presenting the computation times for both the formulations, here we use the five-target instance to illustrate the estimation of the number of allowed visits in a walk.

3) Estimation of $\bar{k}$: Suppose the fuel capacity of the UAV is specified in terms of a limit on its travel time; say the UAV has fuel to travel for a maximum of 80 units of time. Here, we explain the steps in the estimation procedure for $\bar{k}$ in the appendix through an example. First, one can arbitrarily choose an initial estimate for $\bar{k}$. Even though the targets are not equidistant, we start with an estimate that is obtained based on the average travel times between the targets. In the five-target instance, the average travel time between the targets is 9.54 units. Therefore, we consider an initial estimate of $\lfloor \frac{80}{9.54} \rfloor = 8$ visits for $\bar{k}$.

Next, we compute an optimal walk with eight visits. Here, the walk was computed to be (1,2,5,3,5,3,4,2,1) and has a travel time of 39.18 units. Because this value is below the allowed 80 units of travel time, one can keep increase the estimate of $\bar{k}$ till the travel time of its corresponding optimal walk exceeds 80 units. Here, an optimal walk with 14 visits ($\mathcal{W}_0^*(7) \circ \mathcal{W}_0^*(7)$) has a travel time of 76.12 units, whereas an optimal walk with 15 visits ($\mathcal{W}_0^*(5) \circ \mathcal{W}_0^*(5) \circ \mathcal{W}_0^*(5)$) has a travel time of 110.92 units. Hence, based on the proposed estimation procedure, we set $\bar{k}$ to 14 visits. Note that this value is significantly different from the initial estimate which starts from the average travel times between the targets. The reason for this is that the estimation procedure accounts for visits between nearby targets in the optimal walk. This suggests that the usage of $\bar{k}$ as a proxy for the battery charge or travel time is not restricted to specific target configurations.

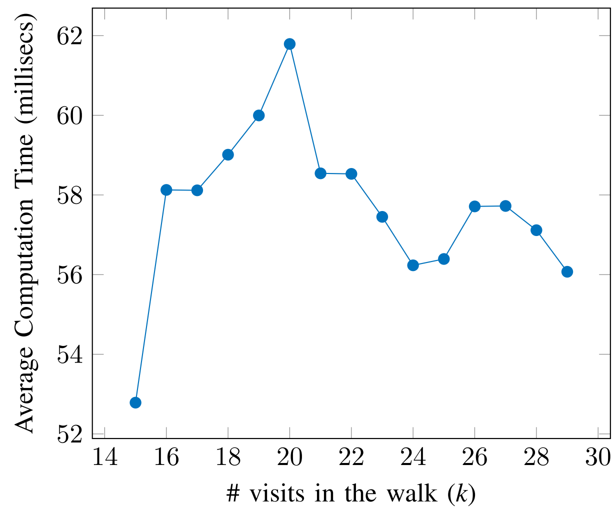


Fig. 18. Computation times with the ILP formulation for 50 instances, each containing 15 targets, against the number of visits in the walk.20

Nonetheless, the estimation procedure can be affected by multiple solutions. For a given number of visits, one can construct multiple optimal walks. For example, when $k = 19$, an optimal walk can be constructed as either $\mathcal{W}_0^*(7) \circ \mathcal{SW}_0(6) \circ \mathcal{SW}_0(6)$ or $\mathcal{W}_0^*(7) \circ \mathcal{W}_0^*(7) \circ \mathcal{SW}_0(5)$. While both the walks have a revisit time of 38.058 units, their travel times are 114.14 units and 113.30 units, respectively. The former walk is not allowed when the limit on the travel time is say 114 units.

C. Computation Time With the ILP Formulation

We demonstrate the strength of the results developed in Sections IV and V, through the average time required for computing optimal solutions with the ILP formulation (given in the Appendix) for large instances with up to 50 targets. For each of $n = 4, 5, 6, 8, 10, 15, 20, 50$, 50 instances were generated as discussed at the beginning of the section. For every instance, optimal walks and optimal revisit times were computed from n to $2n - 1$ visits with the ILP formulation using the JuMP package in Julia. For every set of instances with the same number of targets, the time required to compute optimal solutions were noted and the average of these computation times were plotted against the number of visits in the walks. Sample plots for instances with $n = 15$, $n = 20$, $n = 30$, and $n = 50$ are shown in Figs. 18–21 respectively.

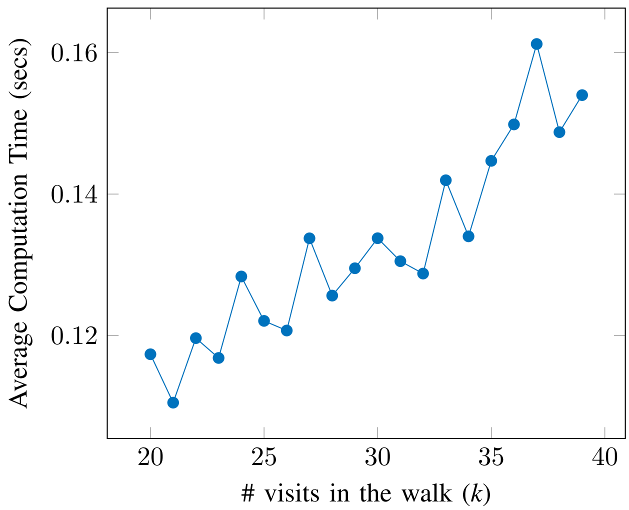


Fig. 19. Computation times with the ILP formulation for 50 instances, each containing 20 targets, against the number of visits in the walk.21

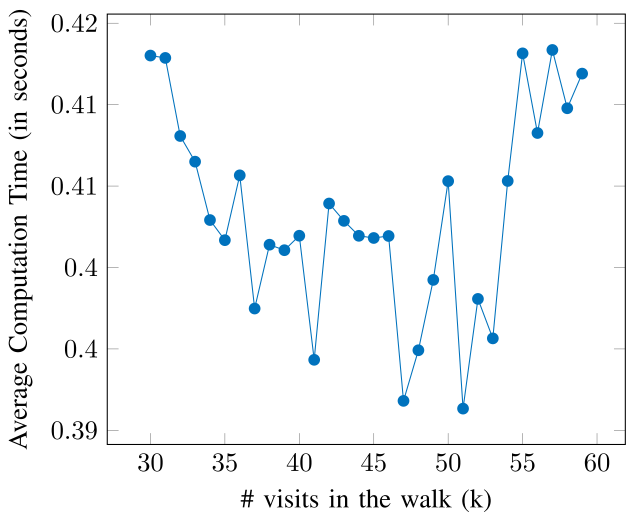


Fig. 20. Average computation times with the ILP formulation for 50 instances, each containing 30 targets, against the number of visits in the walk.22

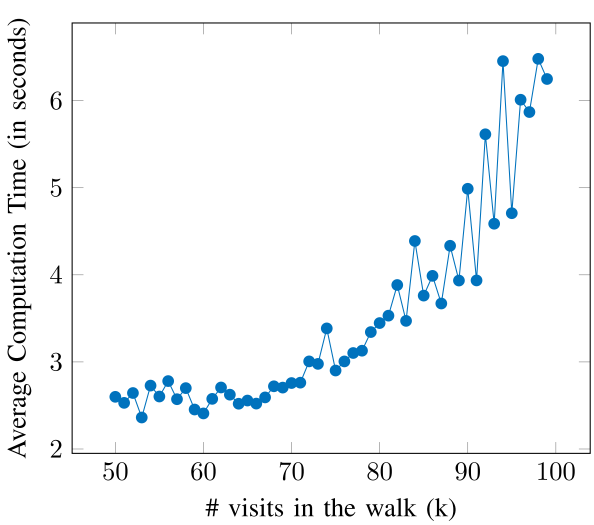


Fig. 21. Average computation times with the ILP formulation for 50 instances, each containing 50 targets, against the number of visits in the walk.23

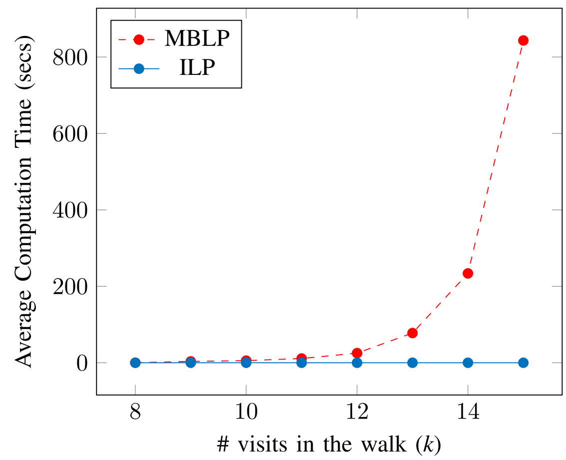


Fig. 22. Comparison of the average computation times for using the MBLP and ILP formulations on 50 instances, each containing 8 targets.18

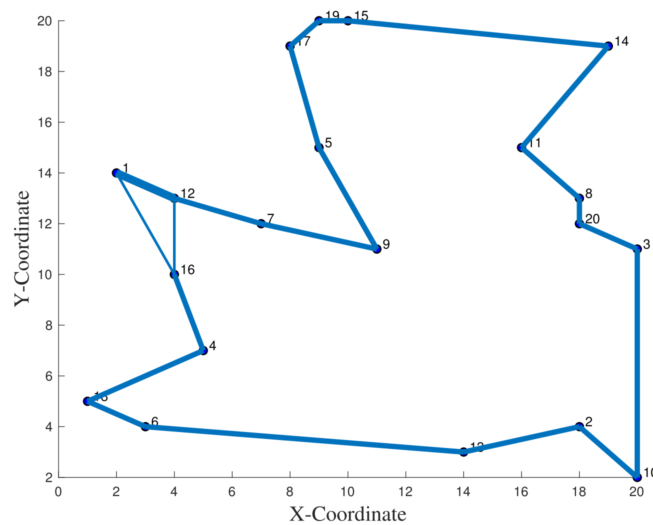


Fig. 23. Optimal walk with 410 visits to 20 targets; here, the width of an edge is proportional to number of times the edge is traversed in the optimal walk.19

First, we compare the computation times for the ILP and the MBLP formulation. For this comparison, we use instances with $n = 8$, which are neither too small for demonstrating the strength of the ILP, nor too high for solving the MBLP. For $8 \leq k \leq 15$, the average computation times for both the formulations are presented in Fig. 22. From the figure, it can clearly be seen that the average time required for solving the ILP is significantly lower compared to that for solving the MBLP. While the maximum and mean computation times over all the visits from n to $2n - 1$ were 843.250 and 150.051 s, respectively, for the MBLP, the same for the ILP were merely 0.038 and 0.036 s, respectively.

Next, for large sized instances, the MBLP was interminable even when the number of visits was as small as $n + 1$; for an instance with 20 targets, the time needed for solving the MBLP for merely 21 visits was 165 273 s. The ILP formulation, on the other hand, produced optimal solutions within a fraction of second for instances upto 30 targets. The average computation times with the ILP formulation for instances with $n = 15$, $n = 20$, $n = 30$, $n = 50$ were 0.057, 0.132, 0.403, and 3.511 s, respectively. With such low computation times, optimal walks with large number of visits can be computed instantaneously; Fig. 23 depicts an optimal walk with 410 visits to 20 targets computed within a fraction of a second using the ILP formulation and the constructed procedure discussed in this article.

Finally, the computation times with the MBLP were observed increase with the number of visits in the walk. Although a similar trend was observed with the ILP for instances with $n = 20$ and $n = 50$, the increase was not consistent across all the sets of instances. So, with the ILP, it is not difficult to compute optimal walks with $2n - 1$ visits compared to those with lesser number of visits.

VII. Conclusion

The computationally challenging problem of finding optimal (minimum revisit time) routes for a UAV while accounting for the charge capacity and associated servicing of the UAV was considered and solved in this work. The difficulty of simultaneously modeling the revisit time and fuel capacity constraints was circumvented by considering a discrete surrogate of k visits in each cycle. The intractability of finding k-visit walks with the least revisit time was surmounted by characterizing the structure of optimal routes for different values of k. This structure can be exploited to develop quick solutions to the problem with the help of either the ILP presented in this article or the vast research done on TSP-type problems, thereby making an online implementation viable. One can also develop heuristic/approximation algorithms to solve the problem for small values of k, and then use the obtained solutions to construct routes with larger number of visits that have the same guarantees as those for the smaller number of visits.

Future work will be focused on developing optimal walks for the case with $0 < \Delta < 2c_{\min}$. We will also investigate the variant in which the depot is used only for servicing and not for data collection or drop-off [24]. This variant is useful when there are no constraints on the communication between the UAV and the station at which the operator is located. In this new variant, the depot can also be considered as a target with a low priority/weight that needs to be visited less often. This leads to an obvious transition to the case with arbitrarily weighted targets. One can systematically introduce weighted targets to potentially solve the generic problem of developing optimal walks for the persistent monitoring of arbitrarily weighted targets. Other variants of the problem such as extensions to multiple UAVs and curvature-constrained UAVs can be found in [25] and [26], respectively.

Appendix A
Estimation of $\bar{k}$

A good estimate of $\bar{k}$ requires a knowledge of the actual sequence in which the UAV visits the targets. Here, we use optimal walks corresponding to different number of visits to iteratively estimate $\bar{k}$. Such a method is possible because of the contributions of this article, which allow a quick computation of optimal walks for a given number, k, of visits. The estimation procedure is given as follows.

1) *Initialization:* Initialize, respectively, upper and lower bounds for $\bar{k}$, i.e., $\bar{k}^u = \infty$ and $\bar{k}^l = n$, where n is the number of targets. Pick any $k \geq n + 1$.
2) *Travel time required per walk:* Using the efficient procedure for solving PMP, determine an optimal walk with k visits, and the corresponding travel time; in case an optimal walk is not available, one can use any feasible walk that is intended to be implemented.
3) *Update the bounds:* If the travel time is within the specified bound and the UAV can execute the walk, update the lower bound $\bar{k}^l = k$; else update the upper bound $\bar{k}^u = k$.
4) *Checking for termination:* If $\bar{k}^u - \bar{k}^l \neq 1$, then pick any k lying between $\bar{k}^l$ and $\bar{k}^u$ and continue to step 2; otherwise, set $\bar{k} = k$ and terminate.

Appendix B
Integer Linear Program (ILP) Formulation

An ILP is presented for computing the optimal walk for the PMP with k visits when $n \leq k \leq 2n - 1$. These solutions can be extended to construct optimal walks for $k \geq 2n$ when $\Delta = 0$ and for $k \geq n^2 + n$ when $\Delta \geq 2c_{\min}$ using the results from Sections IV and V. The formulation utilizes the fact that the travel time of a walk is equivalent to its revisit time, for the case $n \leq k \leq 2n - 1$. Hence, the objective of the formulation is to find a walk with the least travel time. The formulation is a generalization of a commonly used ILP formulation [27] for the TSP.

The benefits of this formulation are twofold: 1) constraints modeling the revisit time (refer to the MBLP formulation of [21]) that contribute overly to the computational difficulty can be circumvented; 2) an optimal route with the least travel time can be computed relatively *quickly*, due to the extensive research done on TSPs.

This formulation is interwoven with the graphical representation of the walk, in contrast to the cyclic representation used for the MBLP formulation. Consider a directed graph G with no self-loops, such that its vertices represent the targets, edges represent the paths connecting the targets, and edge weights represent the travel times of these paths. Then, the decision variables of the formulation are the number of times each edge is utilized in a walk. A feasible solution is a subgraph satisfying the degree and connectivity requirements of a walk. The number of edges in the subgraph is equal to the number of visits in the walk. An ILP for finding an optimal walk with $n \leq k \leq 2n - 1$ visits can be formulated as follows.

Sets:

$\mathcal{T}$: Set of all the targets to be monitored (set of vertices of G).

$\mathcal{E}$: Set of all edges in the graph G connecting any two targets in $\mathcal{T}$.

$\delta^-(i)$: Set of all incoming edges to target i $\forall i \in \mathcal{T}$, i.e., $\delta^-(i) = \{(j, i) \in \mathcal{E} : j \in \mathcal{T} \setminus i\}$.

$\delta^+(i)$: Set of all outgoing edges from target i $\forall i \in \mathcal{T}$, i.e., $\delta^+(i) = \{(i, j) \in \mathcal{E} : j \in \mathcal{T} \setminus i\}$.

Data:

c_{ij}: Time taken by the UAV to travel along an edge $(i, j) \in \mathcal{E}$ (i.e., from target i to target j).

Decision Variables:

x_{ij}: Integer variable (nonnegative) indicating the number of times edge $(i, j) \in \mathcal{E}$ appears in the walk.

Objective:

The objective here is to minimize the total travel time of the walk, that is,

$$\min \sum_{(i,j)\in\mathcal{E}} c_{ij}x_{ij}. \tag{1}$$

Constraints:

To ensure the resulting subgraph is a feasible walk, degree and connectivity constraints are imposed. First and foremost, the number of edges in the solution must be equal to the number of visits in the walk.

1) Total number of edges in the walk is k

$$\sum_{e\in\mathcal{E}} x_e = k. \tag{2}$$

Next, each vehicle must be monitored/visited at least once. Hence, the number of incoming edges to a target $i \in \mathcal{T}$ must be at least 1.

1) Each target must be visited at least once

$$\sum_{e\in\delta^-(i)} x_e \geq 1 \quad \forall i \in \mathcal{T}. \tag{3}$$

Whenever the vehicle enters (travels to) a target, it must also leave the target. In other words, the number of incoming and outgoing edges must be equal at every target $i \in \mathcal{T}$.

1) In-degree must be equal to the out-degree at every target

$$\sum_{e\in\delta^-(i)} x_e = \sum_{e\in\delta^+(i)} x_e \quad \forall i \in \mathcal{T}. \tag{4}$$

As the vehicle visits each target at least once, every subset of targets must have at least one outgoing edge. In simple terms, for the resulting subgraph to be a feasible walk, it must be connected. These constraints can be modeled as follows, for every subset S of $\mathcal{T}$

1)

$$\sum_{e\in\{(i,j):i\in S, j\in\mathcal{T}\setminus S\}} x_e \geq 1 \quad \forall S \subset \mathcal{T}, S \neq \phi. \tag{5}$$

The objective function (1), together with constraints (2)–(5), constitutes the ILP.

Commercial solvers such as CPLEX and Gurobi can be employed to solve the ILP. However, it is to be noted that the number of connectivity constraints is exponential ($2^n - 2$); including all such constraints renders the optimization problem intractable. To circumvent this issue, we implement these constraints in a "lazy" fashion. That is, we first solve a relaxed subproblem by disregarding the connectivity constraints. If the solution to the subproblem is a connected subgraph, then it is also the desired optimal walk. If the resulting subgraph is disconnected, then the subproblem is resolved by adding a connectivity constraint that is violated. This process is repeated until the resulting solution is a connected subgraph. Note that this is a common way of handling subtour elimination constraints for TSP type problems [27]. A violated connectivity constraint is identified by find a minimum weighted cut in the obtained subgraph. If the weight of the minimum cut is zero, then the subgraph is disconnected (note that the travel times are positive). The lazy addition of the violated cuts to the problem, can be easily implemented using the "lazycut" feature in JuMP (a Julia package).

A. *Obtaining a Feasible Walk From the ILP Formulation*

As discussed earlier, for $n \leq k \leq 2n - 1$, the travel time of a walk is equal to its revisit time. Therefore, any sequence of visits containing exactly the same edges as in the obtained connected subgraph, is an optimal walk. Moreover, it is known that at least one target is visited exactly once in this walk. Such a target has exactly one incoming and one outgoing edge in the subgraph. Let target i be visited exactly once. Then, one can always find a sequence of visits such that the vehicle starts from target i, traverses through all the directed edges specified by the subgraph, and returns to target i only at the end. Such a sequence of visits is an optimal walk with k visits, and can be depicted using the cyclic representation. As the starting target for this walk is i, one can simply perform a cyclic permutation to obtain a desired starting target. Note that a cyclic permutation simply changes the starting target in the walk, but retains the sequence of visits in the walk. Therefore, performing a cyclic permutation on a walk does not change its revisit time.

Appendix C
Challenges in Planning a Large Single-Cycle Walk

Suppose that one is considering a mission in which the total duration is known *a priori*. Then, assuming that the UAV is recharged after every k visits, the total number of servicing cycles in the mission is also known; let this number be m. In this case, one must determine an optimal walk with mk visits, such that the UAV returns to the depot after every k visits for servicing. This problem is not only computationally challenging but also involves certain modeling issues that need to be addressed.

When planning a large single-cycle walk, in addition to the time between visits to the targets, one must also consider the following: time elapsed from the depot to the first visit to a target and the time elapsed from the last visit to a target to the depot. Without bounds on these times, some of the targets might be left unmonitored for long time periods. For example, an optimal walk could take the following form: $\mathcal{W} = (1, 2, 4, 1, 2, 3, 4, 3, 1, 2, 4, 1)$. It can be observed that target 3 is only visited in the midst of the walk. Since the time from the beginning of the mission to the first visit to target 3, and the time from the second visit to target 3 to the end of the mission are not bounded, there is no incentive for revisiting target 3 in the walk. This defeats the purpose of uniformly monitoring the equally weighted targets.

There are multiple ways these times can be accounted for as follows.

1) One can independently include these times in the set of all time between visits to the targets and minimize the maximum value in the set; the maximum value can be considered as the revisit time of the walk.

2) For each target, one can add these two times and consider it as the time between the last and the first visits to the targets; note that this is in fact the time between visits to the target when the walk is repeated. Therefore, this time naturally fits into the set of all time between visits to the targets. The minimum of this set can be considered as the revisit time of the walk.

Currently, there is no characterization available for optimal walks based on the first model for revisit time; from our experience, we believe these walks are much difficult to characterize compared to those corresponding to the second model. As a result, one must use a formulation similar to the MBLP formulation mentioned in Section I to determine an optimal walk with mk visits. However, as discussed in Section I, this approach is computationally challenging; the computational difficulty increases rapidly with the number of targets and the number of visits in the walk. The limited time available for planning walks prior to the execution of missions does not necessarily permit such expensive computations.

If the second model is chosen for revisit time, one can use the results developed in this article to determine an optimal walk with mk visits without explicitly computing it (assuming mk is large). Suppose, if $mk \geq n^2 - n$, as discussed in the remarks following Corollary 4, it is sufficient to compute an optimal walk with either n visits or $n + 1$ visits. Therefore, one can swiftly determine an optimal walk with large number of visits.

References

[1] K. Tiwari, "Multi-robot resource constrained decentralized active sensing for online environment monitoring," Ph.D. dissertation, Japan Advanced Insitute of Science and Technology, Nomi, Japan, Mar. 2018. [Online]. Available: http://hdl.handle.net/10119/15336

[2] S. K. K. Hari, S. Rathinam, S. Darbha, K. Kalyanam, S. G. Manyam, and D. Casbeer, "Efficient computation of optimal UAV routes for persistent monitoring of targets," in *Proc. IEEE Int. Conf. Unmanned Aircraft Syst.*, 2019, pp. 605–614.

[3] A. Machado, G. Ramalho, J.-D. Zucker, and A. Drogoul, "Multi-agent patrolling: An empirical analysis of alternative architectures," in *Proc. Int. Workshop Multi-Agent Syst. Agent-Based Simul.* 2002, pp. 155–170.

[4] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Proc. IEEE/WIC/ACM Int. Conf. Intell. Agent Technol.*, 2004, pp. 302–308.

[5] F. Pasqualetti, A. Franchi, and F. Bullo, "On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 592–606, Jun. 2012.

[6] C. G. Cassandras, X. C. Ding, and X. Lin, "An optimal control approach for the persistent monitoring problem," in *Proc. 50th IEEE Conf. Decis. Control Euro. Control Conf.*, 2011, pp. 2907–2912.

[7] K. Kalyanam, S. Manyam, A. V. Moll, D. Casbeer, and M. Pachter, "Scalable and exact MILP methods for UAV persistent visitation problem," in *Proc. 2nd IEEE Conf. Control Technol. Appl.,* 2018, pp. 337–342.

[8] G. Magesh, "Persistent surveillance using a network of MAVs," master's thesis, Aerospace Engineering, Indian Institute of Science, Bangalore, Bangalore, India, 2011.

[9] J. Scherer and B. Rinner, "Persistent multi-UAV surveillance with energy and communication constraints," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, 2016, pp. 1225–1230.

[10] S. G. Manyam, S. Rasmussen, D. W. Casbeer, K. Kalyanam, and S. Manickam, "Multi-UAV routing for persistent intelligence surveillance & reconnaissance missions," in *Proc. Int. Conf. Unmanned aircraft Syst.*, 2017, pp. 573–580.

[11] Y.-W. Wang, Y.-W. Wei, X.-K. Liu, N. Zhou, and C. G. Cassandras, "Optimal persistent monitoring using second-order agents with physical constraints," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3239–3252, Aug. 2019.

[12] J. Las Fargeas, B. Hyun, P. Kabamba, and A. Girard, "Persistent visitation under revisit constraints," in *Proc. Int. Conf. Unmanned Aircraft Syst.*, 2013, pp. 952–957.

[13] E. Tuyishimire, A. Bagula, S. Rekhis, and N. Boudriga, "Cooperative data muling from ground sensors to base stations using UAVs," in *Proc. IEEE Symp. Comput. Commun.*, 2017, pp. 35–41.

[14] H.-M. Ho and J. Ouaknine, "The cyclic-routing UAV problem is PSPACE-complete," in *Proc. Int. Conf. Found. Softw. Sci. Comput. Struct.*, 2015, pp. 328–342.

[15] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of multiple UAVs for persistent surveillance: Algorithm and flight test results," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 5, pp. 1236–1251, Sep. 2012.

[16] N. Drucker, H.-M. Ho, J. Ouaknine, M. Penn, and O. Strichman, "Cyclic-routing of unmanned aerial vehicles," *J. Comput. Syst. Sci.*, vol. 103, pp. 18–45, 2019.

[17] S. Alamdari, E. Fata, and S. L. Smith, "Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 138–154, 2014.

[18] J. R. Alfonsin, *The Diophantine Frobenius Problem.* London, U.K., Oxford Univ. Press, 2005.

[19] D. H. Mohammed, "Generalised Frobenius numbers: Geometry of upper bounds, frobenius graphs and exact formulas for arithmetic sequences," Ph.D. dissertation, School of Mathematics, Cardiff University, Cardiff, U.K., Feb. 2017. [Online]. Available: https://www.overleaf.com/project/5c86d3f153391d448247bd01

[20] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Rev.*, vol. 59, no. 1, pp. 65–98, 2017.

[21] S. K. K. Hari, S. Rathinam, S. Darbha, K. Kalyanam, S. G. Manyam, and D. Casbeer, "Persistent monitoring of dynamically changing environments using an unmanned vehicle," 2018, *arXiv:1808.02545*.

[22] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2019. [Online]. Available: http://www.gurobi.com

[23] I. Dunning, J. Huchette, and M. Lubin, "Jump: A modeling language for mathematical optimization," *SIAM Rev.*, vol. 59, no. 2, pp. 295–320, 2017.

[24] S. K. K. Hari, "Optimal routing of unmanned vehicles in persistent monitoring missions," Ph.D. dissertation, Mechanical Engineering, Texas A & M University, College Station, TX, USA, Dec. 2019.

[25] S. K. K. Hari, S. Rathinam, S. Darbha, K. Kalyanam, S. G. Manyam, and D. Casbeer, "Bounding algorithms for persistent monitoring of targets using unmanned vehicles," in *Proc. IEEE Int. Conf. Unmanned Aircraft Syst.,* 2019, pp. 615–621.

[26] S. K. K. Hari, S. Rathinam, S. Darbha, K. Kalyanam, S. G. Manyam, and D. Casbeer–, "The generalized persistent monitoring problem," in *Proc. Amer. Control Conf.* 2019, pp. 2783–2788.

[27] A. Claus, "A new formulation for the travelling salesman problem," *SIAM J. Algebraic Discr. Methods*, vol. 5, no. 1, pp. 21–25, 1984.

Sai Krishna Kanth Hari received the B.Tech. (Hons.) degree in civil engineering from the Indian Institute of Technology, Kharagpur, Kharagpur, India, in 2014, and the M.S. and Ph.D. degrees in mechanical engineering from the Texas A&M University, College Station, College Station, TX, USA, in 2016 and 2019, respectively.

He is currently a Postdoctoral Research Associate with the Los Alamos National Laboratory, Los Alamos, NM, USA. His research interests include route planning and localization of autonomous vehicles, and control and optimization of infrastructure networks.

Dr. Hari was awarded the Emil Beuhler Aerodynamic Analog Fellowship by the Texas A&M University in 2017 and 2018.

Sivakumar Rathinam (Senior Member, IEEE) received the Ph.D. degree in civil systems engineering from the University of California, Berkeley, Berkeley, CA, USA, in 2007.

He is currently an Associate Professor with the Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA. He worked as a Research Scientist with the NASA Ames Research Center, Mountain View, CA, from 2007 to 2008. He has been with the Texas A&M University since 2009. His research interests broadly include autonomous vehicles, motion planning, optimization, vision based control, and air traffic control.

Dr. Rathinam was the recipient of faculty fellowships from ONR and AFRL, and the best paper award in the 2015 International Conference on Unmanned Aircraft Systems. He is currently an Associate Editor for the IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Intelligent Transportation Systems, IEEE Robotics and Automation Letters, and the *ASME Journal on Dynamic Systems, Measurement, and Control*.

Swaroop Darbha (Fellow, IEEE) received the B.Tech. degree from the Indian Institute of Technology, Madras, Chennai, India, in 1989, and the M.S. and Ph.D. degrees from the University of California at Berkeley, Berkeley, CA, USA in 1992 and 1994 respectively, all in mechanical engineering.

He was a Postdoctoral Researcher with the California PATH program from 1995 to 1996. Since 1997, he has been with the faculty of Mechanical Engineering, Texas A&M University, College Station, TX, USA, where he is currently the Gulf Oil/Thomas A. Dietz Professor. His current research interests lie in the development of vehicle control and diagnostic systems for autonomous ground vehicles, development of planning, control and resource allocation algorithms for a collection of unmanned aerial vehicles.

Dr. Darbha is a Fellow of ASME.

Krishna Kalyanam (Senior Member, IEEE) received the B.Tech. degree from the Indian Institute of Technology, Madras, Chennai, India, in 2000, and the M.S. and Ph.D. degrees from the University of California at Los Angeles, Los Angeles, CA, USA, in 2003 and 2005, respectively, all in mechanical engineering.

He was a Research Engineer with GE Global Research, Niskayuna, NY, USA. from 2005 to 2009. From 2011 to 2018, he was an on-site Contractor/Scientist (InfoSciTex Corp.) with the Autonomous Control Branch, Air Force Research Laboratory (AFRL), Wright-Patterson Air Force Base, OH, USA. He is currently with the Palo Alto Research Center, Palo Alto, CA, USA (Xerox PARC). His research interests include autonomous systems, optimization, and stochastic resource allocation.

Dr. Kalyanam was the recipient of a National Research Council (NRC) Research Associateship affiliated with the Air Vehicles Directorate, AFRL, in 2009.

Satyanarayana Gupta Manyam received the B.E degree in mechanical engineering from the Birla Institute of Technology and Science, Pilani, India, the M.S degree in mechanical engineering from State University of New York at Buffalo, Buffalo, NY, USA, and the Ph.D degree in mechanical engineering from Texas A&M University, College Station, TX, USA, in 2015.

He is currently a Research Scientist with Infoscitex corporation, Dayton, OH, USA, and a Contractor for Air-Force Research Laboratories (AFRL), Wright-Patterson Air Force Base. He was a NRC Postdoctoral Research Associate with Controls Center of Excellence, AFRL. His main research interests include co-operative routing and control of multiagent systems, optimal path and trajectory planning for autonomous vehicles and approximation algorithms.

Dr. Manyam was the recipient of the Best Paper Award at the International Conference on Unmanned Aircraft Systems, 2015.

David Casbeer (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering Brigham Young University, Provo, UT, USA, in 2003 and 2009, respectively.

He is the team lead for the UAV Cooperative and Intelligent Control Team within the Control Science Center of Excellence, Air Force Research Laboratory's Aerospace Systems Directorate, Wright-Patterson Air Force Base, OH, USA. The UAV team focuses on decision making, planning, and coordination for multiple autonomous UAVs acting and reacting in uncertain and adversarial environments. where he advanced theory describing the statistics of decentralized estimation techniques.

Dr. Casbeer was awarded AFRL's Early Career Award for distinguished foundational research in multiagent control, in 2016. He is currently a Senior Editor for the *Journal of Intelligent and Robotic Systems* and an Editor for the *AIAA Journal of Aerospace Information Systems*. His team was awarded the AFOSR Star Team award for 2018–2020 for excellence in basic research.