Automatica 134 (2021) 109943

Contents lists available at ScienceDirect

Automatica

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

Greedy initialization for distributed persistent monitoring in network systems☆

Shirantha Welikala *, Christos G. Cassandras

Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, USA

ARTICLE INFO

Article history:
Received 10 December 2020
Received in revised form 26 July 2021
Accepted 6 August 2021
Available online xxxx

Keywords:
Multi-agent systems
Optimization
Cooperative Control
Control of networks
Persistent Monitoring
Parametric Control

ABSTRACT

This paper considers the optimal multi-agent persistent monitoring problem defined for a team of agents on a set of nodes (targets) interconnected according to a fixed network topology. The aim is to control this team so as to minimize a measure of overall node state uncertainty evaluated over a finite time interval. A class of distributed threshold-based parametric controllers has been proposed in prior work to control agent dwell times at nodes and next-node destinations by enforcing thresholds on the respective node states. Under such a Threshold Control Policy (TCP), an on-line gradient technique was used to determine optimal threshold values. However, due to the non-convexity of the problem, this approach often leads to a poor local optima highly dependent on the initial thresholds used. To overcome this initialization challenge, we develop a computationally efficient off-line greedy technique based on the asymptotic analysis of the network system. This analysis is then used to generate a high-performing set of initial thresholds. Extensive numerical results show that such initial thresholds are almost immediately (locally) optimal or quickly lead to optimal values. In all cases, they perform significantly better than the locally optimal solutions known to date.

1. Introduction

Persistent monitoring of a dynamically changing environment using a cooperating fleet of mobile agents has many applications across different domains. For example, in smart cities (Rezazadeh & Kia, 2019), transportation systems (Yamashita, Arai, Ota, & Asama, 2003) and manufacturing plants (Liaqat et al., 2019), a team of agents can be used to monitor different regions for congestion, disruptions or any other dynamic events of interest. Further, in a smart grid (Caprari et al., 2010; Fan, Wu, Wang, Cao, & Yang, 2018; Menendez, Auat Cheein, Perez, & Kouro, 2017), a team can be used to inspect power plants and transmission lines to maintain a reliable and safe power system. Additional applications include patrolling (Huynh, Enright, & Frazzoli, 2010), surveillance (Aksaray, Leahy, & Belta, 2015; Maza, Caballero, Capitán, Martínez-De-Dios, & Ollero, 2011), data collecting (Khazaeni & Cassandras, 2018; Smith, Schwager, & Rus, 2011; Zhou, Yu, Andersson, & Cassandras, 2018), coverage (Sun, Welikala, & Cassandras, 2020), particle tracking (Shen & Andersson, 2011) and sensing (Lin & Cassandras, 2015; Trevathan & Johnstone, 2018; Zhou, Cassandras, Yu, & Andersson, 2019).

Monitoring problems in general can be classified based on the nature of the *environment*, *objective* and *dynamics* involved. In particular, based on the nature of the environment, a monitoring problem may have a finite set of "points of interest" (Rezazadeh & Kia, 2019; Welikala & Cassandras, 2020) or lack thereof (Lin & Cassandras, 2015; Maini, Yu, Sujit, & Tokekar, 2018) in the environment to be monitored. Based on the nature of the objective, different monitoring problems can be formulated to optimize event-counts (Yu, Karaman, & Rus, 2015), idle-times (Alamdari, Fata, & Smith, 2014; Hari et al., 2019, 2021), error covariances (Pinto, Andersson, Hendrickx, & Cassandras, 2020) or visibility states (Maini et al., 2018) related to the environment. Finally, based on the nature of the environment dynamics, a monitoring problem can be either deterministic (Song, Liu, Feng, & Xu, 2014; Yu, Schwager, & Rus, 2016; Zhou et al., 2019) or stochastic (Lan & Schwager, 2013; Rezazadeh & Kia, 2019).

The persistent monitoring problem considered in this paper is focused on an n-Dimensional (n-D) environment containing a finite number of points of interest (henceforth called "targets"). The objective of the agent team is to collect information from (i.e., sense) each target to reduce an "uncertainty" metric associated with the target state. In particular, the dynamics of each target's uncertainty metric are deterministic and the global

☆ This work was supported in part by NSF, United States, under grants ECCS-1931600, DMS-1664644, CNS-1645681, by AFOSR, United States under grant FA9550-19-1-0158, by ARPA-E's NEXTCAR, United States program under grant DE-AR0000796 and by the MathWorks, United States. The material in this paper was partially presented at the 21st IFAC World Congress (IFAC 2020), July 12–17, 2020, Berlin, Germany. This paper was recommended for publication in revised form by Associate Editor Bin Zhou under the direction of Editor Ian R. Petersen.

* Corresponding author.
E-mail addresses: shiran27@bu.edu (S. Welikala), cgc@bu.edu (C.G. Cassandras).

https://doi.org/10.1016/j.automatica.2021.109943
0005-1098/

objective is to minimize an overall (average) measure of target uncertainties evaluated over a *finite* horizon by controlling the agent trajectories. Hence, the persistent monitoring problem considered in this paper is a dynamic optimization problem whose solutions are generally not periodic. It is structurally different from the persistent monitoring problems considered in Alamdari et al. (2014) and Hari et al. (2021) where min–max objective functions evaluated over an indefinitely repeated cycle of target visits or an infinite horizon are considered and discrete optimization problems are formulated to derive periodic solutions.

In contrast, the work in Zhou et al. (2018) has addressed a similar persistent monitoring problem but constrained to 1-D environments by formulating it as an optimal control problem and reducing it to a parametric optimization problem. The corresponding optimal parameters have been determined by following a gradient descent process with the gradients evaluated on-line through Infinitesimal Perturbation Analysis (IPA) (Cassandras, Wardi, Panayiotou, & Yao, 2010). In contrast to the 1-D case, finding the solution to the problem of persistent monitoring in 2-D environments is much more complicated (Lin & Cassandras, 2015). However, as a remedy, the works in Khazaeni and Cassandras (2018), Lin and Cassandras (2015) and Pinto et al. (2020) propose to constrain agent trajectories to certain standard families of parametric trajectories (e.g., elliptical, Lissajous and Fourier) and then to use IPA to obtain the optimal agent trajectories within these families. Nevertheless, as pointed out in Zhou et al. (2019), all these problems are non-convex, hence standard gradient-based methods often lead to poor local optima dependent on the initial conditions used.

To overcome the challenges mentioned above, Zhou et al. (2019) exploits the network structure of the monitoring system by adopting a graph topology to abstract targets and feasible inter-target agent trajectories as graph nodes and edges, respectively. Note that this graph abstraction has the added advantage of accounting for physical obstacles that might be present in the environment. In this Persistent Monitoring on Networks (PMN) paradigm, an agent trajectory is fully characterized by a *sequence of targets* to be visited and the corresponding *sequence of dwell times* to be spent at each visited target. Therefore, the controller that optimizes a given objective should yield such a (target, dwell time) sequence for all the agents. Clearly, this optimization problem is significantly more complicated than the NP-hard traveling salesman problem (TSP) (Bektas, 2006) which only involves finding an optimal sequence of nodes to visit.

As an alternative, Welikala and Cassandras (2021) uses a Receding Horizon Control (RHC) technique that requires each agent to repeatedly solve a smaller optimization problem to determine its optimal trajectory using only the local information available to it. This on-line control approach has the advantage of being distributed and parameter-free; on the other hand, it cannot exploit any global information regarding the underlying network structure. In contrast, the parametric approach taken in Zhou et al. (2019) uses a class of controllers characterized by threshold parameters which can be optimized in an on-line distributed manner using gradient descent. However, due to the non-convexity of the associated objective function with respect to these thresholds, this gradient-based approach often converges to a poor local optimum that is highly dependent on the initial thresholds which in Zhou et al. (2019) are generated randomly. Nevertheless, the solutions obtained by this parametric controller may be drastically improved by an *initial off-line step* in which the global information available regarding the underlying network structure is exploited to determine a set of high-performing thresholds used to initialize the IPA-based on-line gradient descent process. This process subsequently converges to an improved set of (still locally optimal) thresholds. Our contribution in this paper can be seen more broadly as a systematic approach to select effective initial conditions for gradient-based methods that solve non-convex optimization problems pertaining to a large class of dynamic multi-agent systems beyond persistent monitoring. In particular, this is accomplished by analyzing the asymptotic behavior of such systems and using the resulting optimal control policies to initialize a parametric class of controllers.

For the PMN systems considered in this paper, our contributions include: (i) The asymptotic analysis of single-agent PMN systems with the agent constrained to follow a periodic and non-overlapping sequence of targets (also called “target-cycle”), (ii) A graph partitioning process that enables the extension of this analysis to the deployment of multiple agents, and (iii) A computationally efficient, off-line technique that constructs a high-performing set of thresholds for PMN problems. As shown through extensive simulation results, the initial thresholds provided by this initialization technique are often immediately optimal (still local). Thus, in such cases, an effective initialization eliminates the need for any subsequent gradient descent process.

The paper is organized as follows. Section 2 provides the problem formulation and reviews the threshold-based control policy (TCP) proposed in Zhou et al. (2019). Section 3 includes the asymptotic analysis and a candidate threshold initialization technique, assuming the underlying PMN problem is single-agent and the network is sufficiently dense. Next, Section 4 generalizes the asymptotic analysis and the threshold initialization technique proposed in Section 3 to any network (still assuming a single-agent PMN scenario). Subsequently, Section 5 further generalizes the proposed threshold initialization technique to multi-agent systems. Section 6 presents several numerical examples and performance comparisons with respect to the solution in Zhou et al. (2019). Finally, Section 7 concludes the paper.

2. Problem formulation

We consider an n-dimensional mission space with M targets in the set $\mathcal{T} = \{1, 2, \ldots, M\}$ and N agents in the set $\mathcal{A} = \{1, 2, \ldots, N\}$. Each target $i \in \mathcal{T}$ is located at a fixed position $X_i \in \mathbb{R}^n$ and each agent $a \in \mathcal{A}$ is allowed to move in the mission space where its trajectory is denoted by $\{s_a(t) \in \mathbb{R}^n, t \geq 0\}$. As proposed in Zhou et al. (2019), we embed a directed network topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to this mission space such that the graph vertices represent the targets (i.e., $\mathcal{V} = \mathcal{T}$) and the graph edges represent the inter-target trajectory segments available for agents to travel (i.e., $\mathcal{E} = \subseteq \{(i, j) : i, j \in \mathcal{V}\}$).

In particular, the shape of each trajectory segment $(i, j) \in \mathcal{E}$ can be considered as a result of a lower level optimal control problem which aims to minimize the travel time that an agent takes to go from target i to target j while accounting for potential constraints in the mission space and agent dynamics. For the purpose of this paper, we assume each trajectory segment $(i, j) \in \mathcal{E}$ to have a fixed such optimal travel time value $\rho_{ij} \in \mathbb{R}_{\geq 0}$. Based on $\mathcal{E}$, the *neighbor set* $\mathcal{N}_i$ of target $i \in \mathcal{V}$ is defined as $\mathcal{N}_i \triangleq \{j : (i, j) \in \mathcal{E}\}$. Note also that the target locations $\{X_i : i \in \mathcal{V}\}$, initial agent locations $\{s_a(0) : a \in \mathcal{A}\}$ and travel time values $\{\rho_{ij} : (i, j) \in \mathcal{E}\}$ are prespecified.

Target model. As in Zhou et al. (2019), each target $i \in \mathcal{V}$ has an associated *uncertainty state* $R_i(t) \in \mathbb{R}$ which follows the dynamics:

$$\dot{R}_i(t) = \begin{cases} 0 & \text{if } R_i(t) = 0 \text{ and } A_i \leq B_i N_i(t), \\ A_i - B_i N_i(t) & \text{otherwise,} \end{cases} \tag{1}$$

where $R_i(0)$ is prespecified, $A_i \in \mathbb{R}_{\geq 0}$ is the uncertainty growth rate, $B_i \in \mathbb{R}_{>0}$ is the uncertainty removal rate by an agent and $N_i(t) = \sum_{a=1}^{N} \mathbf{1}\{s_a(t) = X_i\}$ is the number of agents present at

target i at time t ($\mathbf{1}\{\cdot\}$ is the indicator function). In simple terms, (i) $R_i(t)$ increases at a rate A_i when no agent is visiting it, (ii) $R_i(t)$ decreases at a rate $B_iN_i(t) - A_i$ when $N_i(t) > 0$, and (iii) $R_i(t) \geq 0$, $\forall t \geq 0$. The work in Zhou et al. (2019) points out an attractive queueing system interpretation of this target uncertainty model where each target i is viewed as a node in a queueing network with its A_i and $B_iN_i(t)$ respectively representing the arrival rate and the controllable service rate.

Note that in (1), the proportionality between the target uncertainty decrease rate and $N_i(t)$ allows one to inject stochasticity into the problem setup (if needed, through randomizing $N_i(t)$). Moreover, as we will see in the sequel, the proposed solution in this paper motivates agent behaviors that result in $N_i(t) \leq 2, \forall i, t$. Thus, the aforementioned proportionality allows (1) to apply to any of the applications mentioned in the Introduction).

Agent model. In some persistent monitoring models (Pinto et al., 2020; Zhou et al., 2018), each agent $a \in \mathscr{A}$ is assumed to have a finite sensing range $r_a > 0$ that allows it to decrease a target uncertainty $R_i(t)$, $i \in \mathscr{V}$ whenever $\|s_a(t) - X_i\| \leq r_a$. However, we follow the approach used in Zhou et al. (2019) where the condition $\|s_a(t) - X_i\| \leq r_a$ is replaced by $\mathbf{1}\{s_a(t) = X_i\}$ and the role of the joint detection probability of a target i by the agents is replaced by $N_i(t)$.

Objective function. The objective of this persistent monitoring system is to control the team of agents so as to minimize a measure of *mean system uncertainty* J_T where

$$J_T = \frac{1}{T}\int_0^T \sum_{i=1}^{M} R_i(t)dt. \tag{2}$$

Based on the target dynamics (1) and agent sensing capabilities assumed, to minimize the objective J_T (2) it is intuitive that each agent has to *dwell* (i.e., remain stationary) only at targets that it visits in its trajectory. Moreover, based on the embedded target topology $\mathscr{G}$ that constrains the agent motion, it is clear that when an agent $a \in \mathscr{A}$ leaves a target $i \in \mathscr{V}$ its next target would be some target $j \in \mathscr{N}_i$ that is only reachable by *traveling* on edge $(i, j) \in \mathscr{E}$ for a time duration of ρ_{ij}. In essence, this *dwell-travel* approach aims to minimize the agent time spent outside of targets – which is analogous to minimizing the idle time of servers in a queueing network.

Each time an agent $a \in \mathscr{A}$ arrives at a target $i \in \mathscr{V}$, it has to determine a *dwell time* $\tau_i^a \in \mathbb{R}_{\geq 0}$ and a *next visit* target $v_i^a \in \mathscr{N}_i$. Therefore, for the set of agents, the optimal control solution that minimizes the objective J_T takes the form of a set of optimal dwelling time and next visit target sequences.

Determining such an optimal solution is a challenging task even for the simplest PMN problem configurations due to the nature of the involved search space. This is also clear from comparing the PMN problem to the NP-hard TSP problem – which is much simpler as it only requires finding an optimal sequence of targets to visit – yet still has no known global solutions or performance bound guarantees (except for special cases Held & Karp, 1971).

Threshold-based control policy. To address this challenge, we adopt the threshold-based control policy (TCP) proposed in Zhou et al. (2019). In particular, under this TCP, each agent $a \in \mathscr{A}$ makes its decisions by adhering to a set of pre-specified parameters denoted by $\Theta^a \in \mathbb{R}^{M\times M}$ which serve as thresholds on target uncertainties. Note that the $(i, j)^{\text{th}}$ parameter in the Θ^a matrix is denoted as $\theta_{ij}^a \in \mathbb{R}_{\geq 0}$ $\forall i, j \in \mathscr{V}$.

We denote the set of neighbors of a target i that violates their thresholds (i.e., have higher uncertainty values than respective thresholds) with respect to an agent a residing in target i at time t by $\mathscr{N}_i^a(t) \subseteq \mathscr{N}_i$ (also called *active neighbors*) where

$$\mathscr{N}_i^a(t) \triangleq \{j : R_j(t) > \theta_{ij}^a,\ j \in \mathscr{N}_i\}. \tag{3}$$

Assume an agent a arrives at target i at a time $t = t'$. Then, the dwell time τ_i^a to be spent at target i is determined by: (i) the diagonal element θ_{ii}^a based on the *threshold satisfaction* condition $R_i(t) < \theta_{ii}^a$ and (ii) the *active neighbor existence* condition $|\mathscr{N}_i^a(t)| > 0$ at $t = t' + \tau_i^a$ ($|\cdot|$ is the cardinality operator). Subsequently, agent a's next visit target v_i^a is chosen from the set of active neighbors $\mathscr{N}_i^a(t) \subseteq \mathscr{N}_i$ using the off-diagonal thresholds $\{\theta_{iv}^a : v \in \mathscr{N}_i^a(t)\}$ at $t = t' + \tau_i^a$. Formally,

$$\begin{aligned} \tau_i^a &:= \operatorname*{arginf}_{\tau\geq 0} \mathbf{1}\left\{[R_i(t'+\tau) < \theta_{ii}^a] \ \&\ [|\mathscr{N}_i^a(t'+\tau)| > 0]\right\}, \\ v_i^a &:= \operatorname*{argmax}_{v\in\mathscr{N}_i^a(t'+\tau_i^a)} \left\{R_v(t'+\tau_i^a) - \theta_{iv}^a\right\}. \end{aligned} \tag{4}$$

While the first condition in the τ_i^a expression in (4) ensures that agent a will dwell at target i until at least its own uncertainty $R_i(t)$ drops below θ_{ii}^a, the second condition ensures that when agent a is ready to leave target i there will be at least one neighbor $v \in \mathscr{N}_i$ whose uncertainty $R_v(t)$ has increased beyond the threshold θ_{iv}^a. The v_i^a expression in (4) implies that v_i^a is the neighboring target of i chosen from the set $\mathscr{N}_i^a(t' + \tau_i^a) \subseteq \mathscr{N}_i$ with the largest threshold violation. In all, the update equations in (4) define each agent's dwell time and next visit decision sequence under the TCP.

A key advantage of this TCP approach is that, based on (3) and (4), each agent now only needs to use the neighboring target state information. Thus, each agent operates in a *distributed* manner. An example target topology and an agent threshold matrix are shown in Fig. 1. Note that when certain edges are missing in the graph, the respective off-diagonal entries in Θ^a are irrelevant and hence denoted by $\theta_{ij}^a = \infty$.

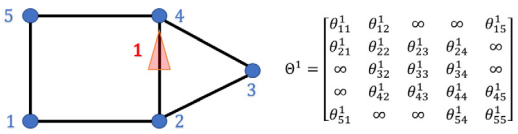


Fig. 1. An example target topology with five targets and one agent with its threshold parameters.

Discrete event system view. Under the TCP mentioned above, the behavior of the PMN system is fully defined by the set of agent decision sequences

$$\mathscr{U}(\Theta) = \{(\tau_{i(l)}^a(\Theta^a), v_{i(l)}^a(\Theta^a)) : l \in \mathbb{Z}_{>0},\ a \in \mathscr{A}\},$$

where $\Theta \in \mathbb{R}^{M\times M\times N}$ is the collection of all agent threshold matrices and $i : \mathbb{Z}_{>0} \to \mathscr{V}$ (in other words, $i(l)$ is the l^{th} target visited by agent a). Following from (4), the PMN system is a discrete event system (DES) (Cassandras & Lafortune, 2010) where the *event set* consists of: (i) agent arrivals and departures at/from targets, (ii) instances where a target uncertainty reaches 0 from above, and (iii) the 'start' and the 'end' events triggered respectively at times $t = 0$ and $t = T$. The sequence of *event times* observed is denoted as $\{t^k : k \in \{0, 1, \ldots, K\}\}$ with $t^0 = 0$ and $t^K = T$.

Since the behavior of the PMN system is dependent on the used TCP Θ, the objective function J_T in (2) is also dependent on Θ. Therefore, within this TCP class of agent controllers, our aim is to determine an optimal TCP (OTCP) Θ^* such that

$$\Theta^* = \operatorname*{argmin}_{\Theta\geq\mathbf{0}} J_T(\Theta) = \frac{1}{T}\sum_{i=1}^{M}\sum_{k=0}^{K}\int_{t^k}^{t^{k+1}} R_i(t)dt. \tag{5}$$

Differentiating the cost $J_T(\Theta)$ w.r.t. parameters Θ yields,

$$\nabla J_T(\Theta) = \frac{1}{T}\sum_{i=1}^{M}\sum_{k=0}^{K}\int_{t^k}^{t^{k+1}} \nabla R_i(t)dt,$$

where $\nabla \equiv \frac{\partial}{\partial \Theta}$. As shown in Zhou et al. (2019), it is easy to see that $\nabla J_T(\Theta)$ reduces to:

$$\nabla J_T(\Theta) = \frac{1}{T}\sum_{i=1}^{M}\sum_{k=0}^{K} \nabla R_i(t^k)(t^{k+1} - t^k). \tag{6}$$

The solution proposed in Zhou et al. (2019) uses Infinitesimal Perturbation Analysis (IPA) (Cassandras et al., 2010) to evaluate the $\nabla R_i(t^k)$ terms (hence $\nabla J_T(\Theta)$) in (6) in an on-line distributed manner. This enables the use of a gradient descent algorithm:

$$\Theta^{(l+1)} = \left[\Theta^{(l)} - \beta^{(l)}\nabla J_T(\Theta^{(l)})\right]^+ \tag{7}$$

to update the TCP Θ iteratively so as to optimize the parameterized objective function $J_T(\Theta)$ ($[\cdot]^+ = \max\{0, \cdot\}$). The step size $\beta^{(l)}$ is selected so that it diminishes with l following the standard conditions $\sum_{l=1}^{\infty}\beta^{(l)} = \infty$ and $\lim_{l\to\infty}\beta^{(l)} = 0$ (Bertsekas, 2016).

Note that we cannot make any assumption regarding the nature of the function $J_T(\Theta)$ (hence, it is generally considered a non-convex function). Therefore, finding globally optimal solutions or performance bound guarantees for (5) is exceptionally challenging without significantly simplifying the PMN problem setup.

Initialization $\Theta^{(0)}$. The work in Zhou et al. (2019) has used a randomly generated set of initial thresholds as $\Theta^{(0)}$ for (7). Due to the non-convexity of the objective function (5), the resulting value of Θ when (7) converges is a local minimum that depends heavily on $\Theta^{(0)}$. Hence, a carefully selected high-performing $\Theta^{(0)}$ can be expected to provide significant improvements over the local minimum obtained from randomly selected $\Theta^{(0)}$. Motivated by this idea, we first investigate the structural and behavioral properties of the underlying PMN system. Then, that knowledge is used to construct a candidate for $\Theta^{(0)}$.

Overview of the PMN solution. For single-agent PMN systems (i.e., with $\mathscr{A} = \{a\}$), the work in Zhou et al. (2019) has proved that it is optimal to make the target uncertainty $R_i(t) = 0$ on each visit of agent a to a target $i \in \mathscr{V}$. This implies that in the OTCP, $\theta_{ii}^a = 0$ (Zhou et al., 2019). Moreover, the empirical results obtained based on Zhou et al. (2019) provide some intuition about high-performing agent behaviors: (i) each agent after a brief initial transient phase converges to a steady state periodic behavior where it cycles across a fixed subset of targets and (ii) agents do not tend to share targets with other agents while in this steady state.

Throughout this paper, we exploit the aforementioned observations to efficiently construct a high-performing (favorable) set of agent trajectories so that it can be translated into a better candidate TCP for $\Theta^{(0)}$ in (7) compared to a randomly generated $\Theta^{(0)}$. It is clear that such a favorable set of agent trajectories takes the form of a non-overlapping set of *target-cycles* on the given graph. This non-overlapping property implies that if we develop a solution for the single-agent PMN problem, it can be extended to multi-agent PMN problems using appropriate graph partitioning and assignment techniques.

Inspired by this discussion, the PMN solution proposed in this paper follows the steps outlined in Alg. 1. Note that we have already discussed Step 6. A key step of Alg. 1 is Step 2, as it requires a technique to find a high-performing agent trajectory (a target-cycle) on a given partition of the graph. In fact, in single-agent PMN problems, we only need to execute Steps 2, 5 and 6 of Alg. 1. Hence, in the following Sections 3 and 4, we assume that only one agent is available (i.e., $N = 1$) and develop a PMN solution by discussing the details of Steps 2 and 5 of Alg. 1. In particular, Section 3 assumes the network to be sufficiently dense and Section 4 relaxes that assumption. The subsequent Section 5 extends the proposed solution to multi-agent problems (i.e., $N > 1$) by discussing the details of Steps 1, 3 and 4 of Alg. 1.

3. Single-agent PMN solution: Bi-triangular networks

In this section, we focus only on single-agent persistent monitoring problems on *sufficiently dense* graphs. More precisely, we consider a given graph to be 'sufficiently dense' if it is *bi-triangular*.

Definition 1. A directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with $|\mathscr{V}| > 3$ is **bi-triangular** if for all $(i, j) \in \mathscr{E}$ there exists $k, l \in \mathscr{V}$ such that $(i, k), (k, j) \in \mathscr{E}$, $(i, l), (l, j) \in \mathscr{E}$, and $k \neq l$.

The network shown in Fig. 17(a) is an example of a bi-triangular graph (a counterexample can be seen in Fig. 14(a)). The following Assumption 1 formally states the conditions assumed in the analysis given in this section. Still, we highlight that this assumption is relaxed in subsequent sections.

Assumption 1. (i) Only one agent is available (i.e., $\mathscr{A} = \{a\}$) and (ii) The given target topology $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is bi-triangular.

Due to Assumption 1, in this section, we first search for a single high-performing target-cycle (i.e., a favorable agent trajectory) in the given graph $\mathscr{G}$ using an iterative greedy scheme. Such a target-cycle is then transformed to a TCP $\Theta^{(0)}$ for the subsequent use in the gradient descent process (7) so as to obtain an OTCP Θ^*. We point out that these three steps correspond to Steps 2, 5 and 6 of Alg. 1, respectively.

We note that this single-agent persistent monitoring setup was introduced in Welikala and Cassandras (2020) without proofs or explicit algorithms. Sections 4 and 5 provide the generalizations to arbitrary networks and multi-agent systems respectively, both not included in Welikala and Cassandras (2020).

3.1. Analysis of an unconstrained target-cycle

We formally define a *target-cycle* as a finite sequence of targets selected from $\mathscr{V}$ of the given graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ such that the corresponding sequence of edges also exists in $\mathscr{E}$. An *unconstrained target-cycle* is a target-cycle with no target on it being repeated. We define $\mathscr{C}$ to be the set of all possible unconstrained target-cycles on the graph $\mathscr{G}$. A generic unconstrained target-cycle in $\mathscr{C}$ is denoted by $\Xi_i = \{i_1, i_2, \ldots, i_m\} \subseteq \mathscr{V}$, where $i_j \in \mathscr{V}$, $\forall j \in \{1, 2, \ldots, m\}$ and $m = |\Xi_i| \leq M$. The corresponding

Algorithm 1 The main steps of the PMN solution.

Input: (i) Target topology (graph) $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, (ii) Set of agents $\mathscr{A} = \{1, 2, \ldots, N\}$ and (iii) Initial conditions: $\{R_i(0) : i \in \mathscr{V}\}$, $\{s_a(0) : a \in \mathscr{A}\}$.
Output: A locally optimal TCP candidate for Θ^* in (5).
1: Partition the given graph $\mathscr{G}$ into N sub-graphs $\{\mathscr{G}_a\}_{a\in\mathscr{A}}$.
2: Find a high-performing agent trajectory in each sub-graph.
3: Refine the sub-graphs along with the agent trajectories.
4: Assign agents to the determined refined agent trajectories (on respective sub-graphs) based on initial agent locations.
5: Obtain the corresponding TCP as $\Theta^{(0)} = \{\Theta^{a(0)} : a \in \mathscr{A}\}$.
6: Use $\Theta^{(0)}$ in (7) and update $\Theta^{(l)}$ using IPA gradients (Zhou et al., 2019).

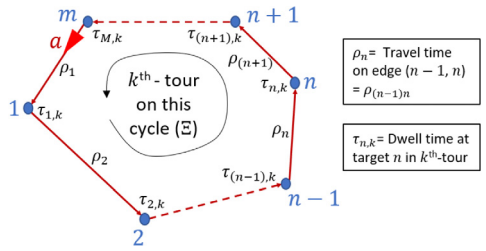


Fig. 2. A generic single-agent unconstrained target-cycle Ξ.

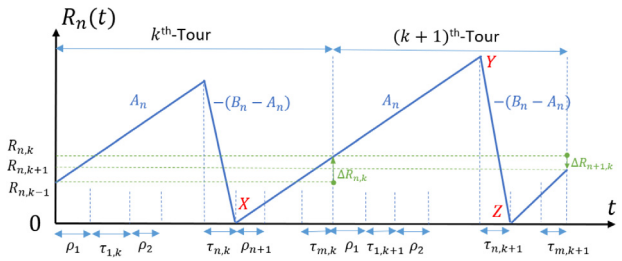


Fig. 3. Variation of target uncertainties during agent tours.

sequence of edges (fully defined by Ξ_i) is denoted by $\xi_i = \{(i_m, i_1), (i_1, i_2), \ldots, (i_{m-1}, i_m)\} \subseteq \mathscr{E}$.

Since we aim to greedily construct a target-cycle which results in a high-performing mean system uncertainty value (i.e., a low J_T in (2)), we need to have an assessment criterion for any given arbitrary target-cycle. Thus, we define the *steady state mean cycle uncertainty* metric $J_{ss}(\Xi_i)$:

$$J_{ss}(\Xi_i) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \sum_{j\in\Xi_i} R_j(t)dt. \tag{8}$$

We now present a computationally efficient off-line method to evaluate $J_{ss}(\Xi_i)$ for any $\Xi_i \in \mathscr{C}$. For notational convenience, we first relabel Ξ_i and its targets as $\Xi = \{1, 2, \ldots, n, n+1, \ldots, m\}$ by omitting the subscript i (see Fig. 2). We then make the following assumption regarding the agent's behavior on a corresponding target-cycle $\Xi \in \mathscr{C}$.

Assumption 2. After visiting a target $n \in \Xi$, the agent will leave it if and only if the target uncertainty R_n reaches zero.

Here, the 'only if' component follows from the aforementioned theoretical result in Zhou et al. (2019): it is optimal to make the target uncertainty $R_n(t) = 0$ whenever the agent visits target $n \in \Xi$. The 'if' component restricts agent decisions by assuming the existence of an active neighbor to target n as soon as $R_n(t) = 0$ occurs in (4). At this point, we recall that our main focus is only on initializing (7) and thus any potential sub-optimalities arsing from the use of Assumption 2 will be compensated by the eventual use of (7).

A *tour* on the target-cycle Ξ (shown in Fig. 2) starts/ends when the agent (i.e., a) leaves the last target m to reach target 1. The dwell time spent on a target $n \in \Xi$ when the agent is in its k^{th} tour on Ξ is denoted as $\tau^a_{n,k}$ and the travel time spent on an edge $(n-1, n) \in \mathscr{E}$ is $\rho_{(n-1)n}$ by definition. Without any ambiguity, we use the notation $\tau_{n,k}$ and ρ_n (with $\rho_1 = \rho_{m1}$) to represent these two quantities respectively. Moreover, target n's uncertainty level at the end of the k^{th} tour is denoted by $R_{n,k}$. Under this notation, the trajectory of the target uncertainty $R_n(t)$ over k^{th} and $(k+1)^{\text{th}}$ tours is shown in Fig. 3. The geometry of the XYZ triangle shown in Fig. 3 can be used to derive the dynamics of target n's dwell time $\tau_{n,k}$ (w.r.t. k) as

$$(B_n - A_n)\tau_{n,k+1} = A_n\Big(\sum_{i=n+1}^{m}\big[\rho_i + \tau_{i,k}\big] + \sum_{i=1}^{n-1}\big[\rho_i + \tau_{i,k+1}\big] + \rho_n\Big). \tag{9}$$

Setting $\alpha_n \triangleq \frac{B_n - A_n}{A_n}$ and $\rho_\Xi \triangleq \sum_{i=1}^m \rho_i$ (the total cycle travel time), the above relationship (9) can be simplified as:

$$-\sum_{i=1}^{n-1}\tau_{i,k+1} + \alpha_n\tau_{n,k+1} = \rho_\Xi + \sum_{i=n+1}^{m}\tau_{i,k}. \tag{10}$$

Note that (10) can be written for all $n \in \Xi$ in a compact form using the vectors $\bar\tau_k = [\tau_{1,k}, \tau_{2,k}, \ldots, \tau_{m,k}]^T$, $\bar\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_m]^T$ and $\bar 1_m = [1, 1, \ldots, 1]^T \in \mathbb{R}^m$, as:

$$\Delta_1\bar\tau_{k+1} = \Delta_2\bar\tau_k + \bar 1_m\rho_\Xi, \tag{11}$$

where $\Delta_2 \in \mathbb{R}^{m\times m}$ is the strictly upper triangular matrix with all non-zero elements being 1 and $\Delta_1 = diag(\bar\alpha) - \Delta_2^T$. The affine linear system expression in (11) describes the evolution of agent dwell times at targets on the target-cycle Ξ over the number of tours completed k. To get an explicit expression for the steady state mean cycle uncertainty $J_{ss}(\Xi)$ defined in (8), we make use of the following three lemmas.

Lemma 1 (*Shermon–Morrison Lemma, Miller, 1981*)**.** *Suppose $A \in \mathbb{R}^{m\times m}$ is an invertible matrix and $u, v \in \mathbb{R}^{m\times 1}$ are vectors. Then, $det(A + uv^T) = (1 + v^TA^{-1}u)det(A)$ and*

$$(1 + v^TA^{-1}u) \neq 0 \iff (A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}.$$

Lemma 2. *When $\sum_{i=1}^m \frac{A_i}{B_i} < 1$, the dynamic system given in (11) has a feasible equilibrium point $\bar\tau_{eq}$ (reached at $k = k_{eq}$),*

$$\bar\tau_{eq} = \left(\frac{\bar\beta}{1 - \bar 1_m^T\bar\beta}\right)\rho_\Xi, \quad \textit{i.e.,}\ \tau_{n,k_{eq}} = \left(\frac{\beta_n}{1 - \sum_{i=1}^m\beta_i}\right)\rho_\Xi, \tag{12}$$

for all $n \in \Xi$ with $\beta_n \triangleq \frac{A_n}{B_n}$ and $\bar\beta = [\beta_1, \beta_2, \ldots, \beta_m]^T$.

Proof. At $k = k_{eq}$, in (11), $\bar\tau_{k+1} = \bar\tau_k = \bar\tau_{eq}$. Therefore, $\bar\tau_{eq} = (\Delta_1 - \Delta_2)^{-1}\bar 1_m\rho_\Xi$. Using $\Delta_1 = diag(\bar\alpha) - \Delta_2^T$ and $diag(\bar 1_m) + \Delta_2^T + \Delta_2 = \bar 1_m\bar 1_m^T$, $\bar\tau_{eq}$ can be simplified as:

$$\bar\tau_{eq} = (diag(\bar\alpha + \bar 1_m) - \bar 1_m\bar 1_m^T)^{-1}\bar 1_m\rho_\Xi. \tag{13}$$

The α_n, β_n expressions give that $(\alpha_n + 1)^{-1} = \beta_n$. Therefore, $(diag(\bar\alpha + \bar 1_m))^{-1} = diag(\bar\beta)$. Also, note that $diag(\bar\beta)\bar 1_m = \bar\beta$ and $I_m \in \mathbb{R}^{m\times m}$ is an identity matrix. Now, using Lemma 1,

$$\bar\tau_{eq} = diag(\bar\beta)\left(I_m + \frac{\bar 1_m\bar 1_m^T diag(\bar\beta)}{1 - \bar 1_m^T\bar\beta}\right)\bar 1_m\rho_\Xi = \left(\frac{\bar\beta}{1 - \bar 1_m^T\bar\beta}\right)\rho_\Xi.$$

Components of $\bar\tau_{eq}$ are non-negative only when $1 - \bar 1_m^T\bar\beta > 0$. Thus, using the definition of $\bar\beta$, we get $\bar 1_m^T\bar\beta = \sum_{i=1}^m \frac{A_i}{B_i} < 1$. ■

To establish the stability properties of $\bar\tau_{eq}$ given by Lemma 2, we next make the following assumption.

Assumption 3. The matrix $\Delta_1^{-1}\Delta_2$ is Schur stable (Bof, Carli, & Schenato, 2018).

Note that the eigenvalues of Δ_2 are located at the origin as it is a strictly upper triangular matrix. Further, the eigenvalues of Δ_1^{-1} are located at $\{\frac{1}{\alpha_i} : i \in \Xi\}$ since Δ_1 is a lower triangular matrix with its diagonal elements being $\{\alpha_i : i \in \Xi\}$. Using the definition of $\alpha_i (= \frac{B_i - A_i}{A_i})$, it is easy to show that $|\frac{1}{\alpha_i}| < 1 \iff 0 \leq \frac{A_i}{B_i} < \frac{1}{2}$, which is a less restrictive condition than $\sum_{i=1}^m \frac{A_i}{B_i} < 1$ required in Lemma 2. Therefore, it seems reasonable to conjecture that the

eigenvalues of $\Delta_1^{-1}\Delta_2$ are located within the unit circle; however, to date, we have not provided a formal proof to the statement in Assumption 3. Nevertheless, since both Δ_1 and Δ_2 matrices are known, the validity of this assumption for any given system (11) can be verified easily.

Lemma 3. *Under Assumption 3, the equilibrium point $\bar{\tau}_{eq}$ given in Lemma 2 for the affine linear system (11) is globally asymptotically stable (i.e., $\lim_{k\to\infty}\bar{\tau}_k = \bar{\tau}_{eq}$, irrespective of $\bar{\tau}_0$).*

Proof. Let $\bar{e}_k = \bar{\tau}_k - \bar{\tau}_{eq}$ as the steady state error. Then, we can write $\bar{e}_{k+1} = \bar{\tau}_{k+1} - \bar{\tau}_{eq}$ and using (11) and Lemma 2,

$$\bar{e}_{k+1} = (\Delta_1^{-1}\Delta_2\bar{\tau}_k + \Delta_1^{-1}\bar{1}_m\rho_\Xi) - (\Delta_1^{-1}\Delta_2\bar{\tau}_{eq} + \Delta_1^{-1}\bar{1}_m\rho_\Xi),$$

so that, $\bar{e}_{k+1} = \Delta_1^{-1}\Delta_2\bar{e}_k$. Therefore, under Assumption 3, all the eigenvalues of $\Delta_1^{-1}\Delta_2$ are within the unit circle. Thus, the equilibrium point $\bar{\tau}_{eq}$ given in (12) of (11) is globally asymptotically stable (Bof et al., 2018). ■

We now present our main theorem regarding the steady state mean cycle uncertainty (8) of the PMN system in Fig. 2.

Theorem 1. *Under Assumptions 2 and 3 with $\sum_{i=1}^m \frac{A_i}{B_i} < 1$, the generic (single-agent) unconstrained target-cycle Ξ in Fig. 2 achieves a steady state mean cycle uncertainty value (i.e., (8)),*

$$J_{ss}(\Xi) = \frac{1}{2}(\bar{B} - \bar{A})^T\bar{\tau}_{eq}, \tag{14}$$

where $\bar{B} = [B_1, B_2, \ldots, B_m]^T$, $\bar{A} = [A_1, A_2, \ldots, A_m]^T$, and $\bar{\tau}_{eq}$ is given in (12).

Proof. Both Lemmas 2 and 3 are applicable under the given conditions. Therefore, using (8) we can write,

$$J_{ss}(\Xi) = \lim_{T\to\infty}\frac{1}{T}\int_0^T\sum_{n=1}^m R_n(t)dt = \frac{1}{T_\Xi}\int_{\partial T_\Xi}\sum_{n=1}^m R_n(t)dt,$$

where $T_\Xi \triangleq \rho_\Xi + \bar{1}_m^T\bar{\tau}_{eq}$ represents the steady state tour duration and ∂T_Ξ is a time period of a tour occurring after achieving steady state. This can be further simplified into

$$J_{ss}(\Xi) = \sum_{n=1}^m\frac{1}{T_\Xi}\int_{\partial T_\Xi} R_n(t)dt.$$

Using the $R_n(t)$ trajectory shown in Fig. 3 note that when equilibrium is achieved (as $T\to\infty$, hence $k\to\infty$), the final tour uncertainties will become stationary (i.e., $R_{n,k} = R_{n,k+1}, \forall n\in\Xi$). Hence the area under the $R_n(t)$ trajectory evaluated over a period T_Ξ becomes equivalent to that of a triangle where the base is T_Ξ and the height is $(B_n - A_n)\tau_{n,\infty}, \forall n\in\Xi$. Therefore,

$$J_{ss}(\Xi) = \sum_{n=1}^m\frac{1}{T_\Xi}\frac{1}{2}T_\Xi(B_n - A_n)\tau_{n,\infty} = \frac{1}{2}(\bar{B}-\bar{A})^T\bar{\tau}_{eq}. \quad ■$$

Theorem 1 enables assessing simple agent trajectories (e.g., the target-cycle in Fig. 2) efficiently and will be used to construct a high-performing target-cycle on the graph $\mathcal{G}$.

3.2. Greedy target-cycle construction

Theorem 1 can be used to identify the best performing (steady state) target-cycle in $\mathscr{C}$ if $|\mathscr{C}|$ is small via exhaustive search evaluating (14) over all Ξ:

$$\Xi^* = \arg\min_{\Xi\in\mathscr{C}} J_{ss}(\Xi). \tag{15}$$

Since this brute-force approach becomes computationally expensive as $|\mathscr{C}|$ grows exponentially with the number of targets or edges, we propose instead a computationally efficient greedy scheme to construct a sub-optimal target-cycle (denoted as $\Xi^\#$) as a candidate for Ξ^* in (15). In this greedy scheme, each iteration search expands a current target-cycle Ξ by adding an unvisited target $i\in\mathscr{V}\backslash\Xi$ to Ξ ($\cdot\backslash\cdot$ is the set subtraction operation). The constructed $\Xi^\#\in\mathscr{C}$ is then transformed to a TCP which is used as $\Theta^{(0)}$ in (7). Therefore, determining the optimal target-cycle Ξ^* is not essential at this stage as opposed to the importance of keeping the overall process of obtaining $\Theta^{(0)}$ efficient.

We now define the finite horizon version of $J_{ss}(\Xi_i)$ in (8) as the *finite horizon mean cycle uncertainty* $J_T(\Xi_i)$, where

$$J_T(\Xi_i) = \frac{1}{T}\int_0^T\sum_{j\in\Xi_i} R_j(t)dt. \tag{16}$$

Note that if $\mathscr{V} = \Xi_i$, this $J_T(\Xi_i)$ metric is equivalent to the mean system uncertainty metric J_T defined in (2).

Contribution of a neglected target. Formally, a *neglected target* is a target that is not visited by any agent during the period $[0, T]$. As our main objective J_T in (2) is evaluated over a finite horizon T, if one or more targets are located remotely compared to the rest of the targets, then neglecting such remote targets might be better than trying to visit them. The following lemma characterizes the contribution of such a neglected target to the main objective J_T in (2).

Lemma 4. *The contribution of a neglected target $i\in\mathscr{V}$ to the mean system uncertainty J_T (defined in (2)) is $\left(R_{i,0} + \frac{A_iT}{2}\right)$.*

Proof. The mean system uncertainty J_T defined in (2) (for the original PMN problem setting with M targets in $\mathscr{V} = \{1, 2, \ldots, M\}$) can be decomposed as $J_T = \frac{1}{T}\int_0^T\sum_{j\in\mathscr{V}\backslash\{i\}} R_j(t)dt + \frac{1}{T}\int_0^T R_i(t)dt$, where the second term represents the contribution of target i to the main objective J_T. Since target i is not being visited by any agent during $t\in[0,T]$ and from (1), $\dot{R}_i(t) = A_i\ \forall t\in[0,T]$. Also note that the initial target uncertainty of target i is $R_i(0) = R_{i,0}$. Therefore, the contribution of target i can be simplified as

$$\frac{1}{T}\int_0^T R_i(t)dt = \frac{1}{T}\int_0^T(R_{i,0} + A_it)dt = \left(R_{i,0} + \frac{A_iT}{2}\right). \quad ■$$

Assumption 4. For any target-cycle $\Xi\in\mathscr{C}$, the difference between the steady state mean cycle uncertainty $J_{ss}(\Xi)$ (defined in (8)) and the finite horizon mean cycle uncertainty $J_T(\Xi)$ (defined in (16)) is bounded by some finite constant $K_e\in\mathbb{R}_{\geq 0}$, i.e., $|J_{ss}(\Xi) - J_T(\Xi)| < K_e$.

The greedy target-cycle construction scheme uses the $J_{ss}(\cdot)$ metric defined in (8) to compare the performance of different target-cycles as it can be evaluated efficiently using Theorem 1. However, since the original objective J_T in (2) is evaluated over a finite horizon T, the $J_T(\cdot)$ metric defined in (16) is more appropriate to evaluate (and compare) different target-cycle performances. The above assumption states that $J_T(\cdot)$ will always lie within $J_{ss}(\cdot)\pm K_e$ and we point out that K_e is small whenever: (i) the steady state tour duration T_Ξ and the finite horizon T is such that $T\gg T_\Xi$, and (ii) the dynamics of the steady state error of (11) are faster (i.e., according to Lemma 3, when $\frac{A_i}{B_i}\ll 1$).

Target-Cycle Expansion Operation (TCEO). Consider the target-cycle $\Xi = \{1, 2, \ldots, m\}$ with its corresponding sequence of edges $\xi = \{(m, 1), (1, 2), \ldots, (m-1, m)\}$. As shown in Fig. 4, in order to expand Ξ so that it includes one more target i chosen from the set of neglected targets $\mathscr{V}\backslash\Xi$, we have to: (i) replace one edge $(n-1, n)$ chosen from ξ with two new consecutive edges $(n-1, i), (i, n)\in\mathscr{E}$ and (ii) insert the neglected target i into Ξ

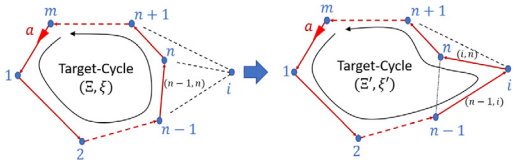


Fig. 4. A basic target-cycle expanding operation (TCEO).

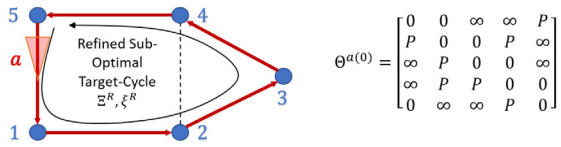


Fig. 5. The generated initial threshold matrix $\Theta^{a(0)}$ (right) for the refined sub-optimal target-cycle Ξ^R (left).

between targets $n-1$ and n. Whenever $|\mathscr{V}\backslash\Xi| > 0$, the existence of a such i and $(n-1, n)$ is guaranteed by the bi-triangularity condition in Assumption 1. Upon executing these two operations, a new (expanded) target-cycle Ξ' (and ξ') is attained as shown in Fig. 4. The following theorem derives the *marginal gain* (denoted as $\Delta J_T(i|\xi, (n-1, n))$) in the main objective J_T in (2) due to such a target-cycle expansion in terms of $J_{ss}(\cdot)$ in (8).

Theorem 2. *Under Assumptions 1, 2 and 4, the marginal gain in the mean system uncertainty J_T (defined in (2)) due to the basic target-cycle expansion operation (shown in Fig. 4) is*

$$\Delta J_T(i|\xi, (n-1, n)) = \left(R_{i,0} + \frac{A_i T}{2}\right) + J_{ss}(\Xi) - J_{ss}(\Xi'). \tag{17}$$

Here, Ξ' is the expanded cycle and $J_{ss}(\cdot)$ is given in Theorem 1. The associated estimation error of this term is $\pm 2K_e$.

Proof. When target i is neglected, Lemma 4 gives the mean system uncertainty as $\left(R_{i,0} + \frac{A_i T}{2}\right) + J_T(\Xi)$. After the target-cycle expansion, the mean system uncertainty is $J_T(\Xi')$ (Note that now $i \in \Xi'$ and $J_T(\cdot)$ is defined in (16)). Therefore, the gain in mean system uncertainty is $\left(R_{i,0} + \frac{A_i T}{2}\right) + J_T(\Xi) - J_T(\Xi')$. Now, adding and subtracting a $(J_{ss}(\Xi) - J_{ss}(\Xi'))$ term and applying Assumption 4 twice (for $J_T(\Xi)$, $J_T(\Xi')$ terms) shows that the above "gain" can be estimated by the marginal gain expression given in (17) (with a tolerance of $\pm 2K_e$). ■

Greedy algorithm. Based on the discussion above and exploiting Theorem 2, Alg. 2 provides a systematic way to construct a candidate (sub-optimal) unconstrained target-cycle $\Xi^\#$ as a solution for (15). Note that Step 1 of Alg. 2 involves $|\mathscr{E}|$ evaluations of $J_{ss}(\cdot)$ (using Theorem 1). In the k^{th} iteration of the subsequent iterative process, if $\mathscr{G}$ is fully connected, Step 3 of Alg. 2 requires $(k+1) \times (|\mathscr{V}| - k - 1)$ evaluations of ΔJ_T (using Theorem 2), where $(k+1)$ is the number of edges in the current target-cycle and $(|\mathscr{V}| - k - 1)$ is the number of remaining neglected targets. Hence the required number of total evaluations of ΔJ_T is clearly much smaller than $|\mathscr{C}|$ in (15).

Algorithm 2 Greedy target-cycle construction for (15).

Input: Graph topology $\mathscr{G} = (\mathscr{V}, \mathscr{E})$.
Output: A sub-optimal target-cycle $\Xi^\#$ (and $\xi^\#$) for (15).
1: Find the target-cycle Ξ in $\mathscr{G}$ with only two targets (i.e., $|\Xi| = 2$) that has the minimum $J_{ss}(\cdot)$ value.
2: **while True do**
3: Find the optimal way (i.e., maximizes the marginal gain ΔJ_T in (17)) to expand Ξ over all possible edges to remove in ξ and targets to add in $\mathscr{V}\backslash\Xi$.
4: If the corresponding marginal gain $\Delta J_T > 0$ execute the expansion (i.e., update Ξ), otherwise, **Break**.
5: **end while**
6: $\Xi^\# := \Xi$; $\xi^\# := \xi$; **Return**;

TSP-inspired target-cycle refinements. Note that the PMN problem is more complicated than the Traveling Salesman Problem (TSP) for several reasons; for instance, in PMN problems, a tour cost value $J_T(\cdot)$, cannot be assigned to individual edges of the topology, but can only be assigned to target-cycles using Theorem 1. However, we can still adopt local search techniques developed for the TSP to further improve the sub-optimal target-cycle $\Xi^\#$ (given by Alg. 2) for the PMN problem. Specifically, we use the conventional 2-Opt and 3-Opt techniques (Blazinskas & Misevicius, 2011; Nilsson, 2003). The main idea behind a step in these two methods is to perturb the shape of $\Xi^\#$ slightly (say into Ξ') and then to check whether $J_{ss}(\Xi') < J_{ss}(\Xi^\#)$. If so, the update $\Xi^\# := \Xi'$ is executed.

3.3. Generating an initial TCP: $\Theta^{(0)}$

Let us denote the final refined sub-optimal target-cycle as Ξ^R (and ξ^R). We now need to transform Ξ^R into a set of TCP parameters to be used as $\Theta^{(0)}$ in (7). Since $\mathscr{A} = \{a\}$ under Assumption 1, we can write $\Theta^{(0)} = \Theta^{a(0)} \in \mathbb{R}^{M \times M}$. Further, note that the TCP values in $\Theta^{(0)}$ should be such that they guide the agent according to Assumption 2 on Ξ^R. To achieve this, we simply follow the steps given in Alg. 3. Note that Step 1 of Alg. 3 ensures that the agent remains at target $i \in \Xi^R$ until $R_i(t) = 0$, whenever it is visited. Moreover, Steps 2, 3 of Alg. 3 ensure that the agent follows the target-cycle Ξ^R (recall that T_{Ξ^R} represents the steady state tour duration on the target-cycle Ξ^R). An example input/output for this algorithm corresponding to the target topology in Fig. 1 is shown in Fig. 5.

Algorithm 3 Generating $\Theta^{a(0)}$ from the target-cycle Ξ^R, ξ^R.

Input: Graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, and the target-cycle Ξ^R, ξ^R.
Output: Initial TCP $\Theta^{a(0)}$ for the use in (7).
1: All the diagonal entries of $\Theta^{a(0)}$ are set to 0.
2: The $(i,j)^{\text{th}}$ entry of $\Theta^{a(0)}$ is set to 0 for all $(i,j) \in \xi^R$.
3: All other (valid/finite) entries of $\Theta^{a(0)}$ are set to a large constant $P \in \mathbb{R}$ where $P > T_{\Xi^R} \max_{i \in \mathscr{V}} A_i$.

4. Single-agent PMN solution: General networks

In this section, we relax the bi-triangularity assumption in Assumption 1(ii) so as to generalize the developed single-agent PMN solution for any network. As a reminder, we will also relax the single-agent assumption in the next section.

When the network $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is sparse, due to the lack of edges in $\mathscr{E}$, the bi-triangularity assumption made in Assumption 1(ii) may no longer hold. As a result of this assumption violation, the iterative target-cycle expansion process (i.e., Alg. 2) might halt prematurely (i.e., while $|\mathscr{V}\backslash\Xi| > 0$) due to the lack of feasible expansions. Two such examples are shown in Fig. 6. One obvious approach to overcome this assumption violation is by inserting (artificial) edges into the network with higher travel time values. However, while such an approach can make Alg. 2 run without halting, the resulting target-cycle $\Xi^\#$ will contain the edges that were artificially introduced, compromising the target-cycle performance $J_{ss}(\Xi^\#)$.

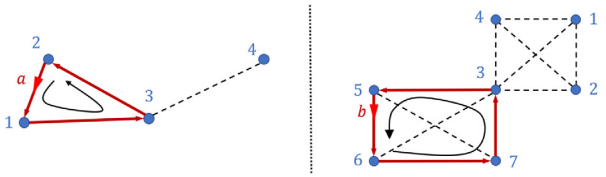


Fig. 6. Two example sparse networks where Alg. 2 has halted prematurely while executing target-cycle expansion iterations.

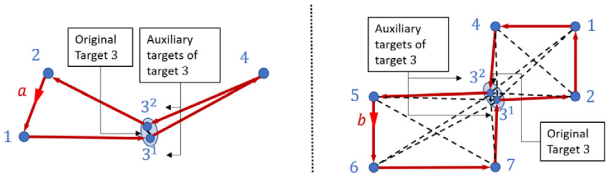


Fig. 7. Converting constrained target-cycles into unconstrained target-cycles with the use of auxiliary targets.

Auxiliary targets. As opposed to introducing artificial edges, we propose to introduce artificial targets (henceforth called *auxiliary targets*) into the network so as to deal with this issue. Unlike artificial edges, an auxiliary target is always associated with a corresponding target in the original network. The physical interpretation of an auxiliary target is provided in the sequel.

Note that if certain targets in a network can be visited more than once, the target-cycle expansion process may not have to be halted due to the lack of edges (sparseness or non-bi-triangularity) in the network. Therefore, we propose to allow targets to be visited more than once during a tour on a target-cycle and we call such target-cycles *constrained target-cycles*. For example, in both networks shown in Fig. 6, if target 3 is allowed to be visited more than once during a tour, we can construct the constrained target-cycles $\bar{\Xi} = \{2, 1, 3, 4, 3\}$ and $\bar{\Xi} = \{6, 7, 3, 2, 1, 4, 3, 5\}$, respectively. Note that we use the notation "$\bar{\cdot}$" to indicate that the target-cycle is constrained (i.e., some elements are being repeated).

To analyze such constrained target-cycles (i.e., to evaluate their $J_{ss}(\cdot)$ values in (8)), we use the previously mentioned concept of *auxiliary targets*. As we will see in the sequel, replacing the repeated targets with a set of carefully chosen auxiliary targets can transform a constrained target-cycle into an *equivalent* unconstrained target-cycle, enabling the application of Theorem 1.

Consider a constrained target-cycle $\bar{\Xi}$ with a target $i \in \bar{\Xi}$ being visited n times during a tour. Then, we first introduce an *auxiliary target pool* $\mathscr{T}_i = \{i^1, i^2, \ldots, i^n\}$ where each auxiliary target $i^j \in \mathscr{T}_i$ can be thought of as an artificial target located in the same physical location of target i (i.e., at X_i in the mission space), but with its own parameters: an uncertainty rate A_i^j and a sensing rate B_i^j (to be defined). Next, we replace the repeated elements of target i in $\bar{\Xi}$ with the elements taken from auxiliary target pool $\mathscr{T}_i$. Then, we repeat this process for all $i \in \bar{\Xi}$ with $|\mathscr{T}_i| > 1$. This results in an unconstrained target-cycle which we denote as Ξ (i.e., without "$\bar{\cdot}$", we follow this notational convention in the rest of this paper).

For example, consider the constrained target-cycles proposed for the graphs in Fig. 6. Now, using the auxiliary target pool $\mathscr{T}_3 = \{3^1, 3^2\}$ (for both graphs), their respective unconstrained target-cycles $\Xi = \{2, 1, 3^1, 4, 3^2\}$ and $\Xi = \{6, 7, 3^1, 2, 1, 4, 3^2, 5\}$ in Fig. 7 can be obtained.

Equivalence criteria. For the analysis of the constrained target-cycles, we enforce the requirement that both the targets in Ξ and $\bar{\Xi}$ should perform/behave in an equivalent manner at steady state. Specifically, we enforce the following *equivalence criteria* between the targets in Ξ and $\bar{\Xi}$.

(1) The dwell time spent at $i^j \in \Xi$ is equal to the dwell time spent at $i \in \bar{\Xi}$ on its j^{th} visit during a tour.
(2) The physical location of $i^j \in \Xi$ is the same as that of $i \in \bar{\Xi}$.
(3) The contribution to the main objective J_T (2) by $\mathscr{T}_i \subset \Xi$ is equal to that of target $i \in \bar{\Xi}$, during a tour.

The first two conditions ensure that the time required to complete a tour (for an agent) is the same for both Ξ and $\bar{\Xi}$. The third condition implies $J_{ss}(\bar{\Xi}) = J_{ss}(\Xi)$. Hence, if the auxiliary target parameters are known, Theorem 1 will yield the value of $J_{ss}(\bar{\Xi})$.

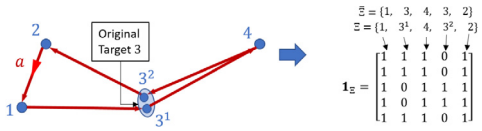


Fig. 8. Sub-cycle unit vectors and sub-cycle matrix (right) for a given constrained target-cycle $\bar{\Xi}$ (left).

Sub-cycles. Notice that each $i^j \in \Xi$ can be assigned a *sub-cycle* denoted by $\Xi_i^j \subset \Xi$ where Ξ_i^j starts with the immediate next target to $i^{j-1} \in \Xi$ and ends with target i^j. Therefore, Ξ can be written as a concatenation of sub-cycles of a target $i \in \bar{\Xi}$, i.e., $\Xi = \bigcup_{i^j \in \mathscr{T}_i} \Xi_i^j$. Also, if for some $i \in \bar{\Xi}$, $|\mathscr{T}_i| = 1$ (i.e., no auxiliary targets), then its sub-cycle is $\Xi_i^1 = \Xi$. For example, for the unconstrained target-cycle Ξ shown in the first graph of Fig. 7, sub-cycles corresponding to $3^1, 3^2 \in \Xi$ are $\Xi_3^1 = \{2, 1, 3^1\}$ and $\Xi_3^2 = \{4, 3^2\}$, respectively.

The *sub-cycle unit vector* of Ξ_i^j is denoted by $\bar{1}_i^j \in \mathbb{R}^{|\Xi|}$ and its n^{th} element is 1 only if the n^{th} element of Ξ belongs to Ξ_i^j. Therefore, if $\bar{1}_{|\Xi|} \in \mathbb{R}^{|\Xi|}$ is a vector of all ones, with respect to target $i \in \bar{\Xi}$, we can write $\bar{1}_{|\Xi|} = \sum_{i^j \in \mathscr{T}_i} \bar{1}_i^j$. Also, if for some $i \in \bar{\Xi}$, $|\mathscr{T}_i| = 1$, then its sub-cycle unit vector is $\bar{1}_i^1 = \bar{1}_{|\Xi|}$.

The *sub-cycle matrix* of Ξ is denoted by $\mathbf{1}_\Xi \in \mathbb{R}^{|\Xi| \times |\bar{\Xi}|}$ and its n^{th} column is the sub-cycle unit vector of the n^{th} element of Ξ. Note that if $\forall i \in \bar{\Xi}, |\mathscr{T}_i| = 1$, then all elements of $\mathbf{1}_\Xi$ will be 1. Fig. 8 shows an example sub-cycle matrix.

4.1. Analysis of constrained target-cycles

We are now ready to analyze a generic constrained target-cycle $\bar{\Xi}$. Throughout this analysis, we will use the constrained target-cycle example shown in Fig. 9 for illustration purposes. Note that in this particular constrained target-cycle, $\bar{\Xi} = \{1, 2, \ldots, n, \ldots, n+m-1, n\}$ and target $n \in \bar{\Xi}$ is visited twice during a tour. Introducing auxiliary targets $\mathscr{T}_n = \{n^1, n^2\}$, $\bar{\Xi}$ can be converted to its equivalent unconstrained version Ξ. The sub-cycles of n^1 and n^2 in Ξ are $\Xi_n^1 = \{1, 2, \ldots, n-1, n^1\}$ and $\Xi_n^2 = \{n+1, n+2, \ldots, n+m-1, n^2\}$, respectively. A *tour* on $\bar{\Xi}$ starts/ends when the agent leaves target n to reach target 1 and we assume the agent behavior on $\bar{\Xi}$ to follow Assumption 2. We label the inter-target travel times on $\bar{\Xi}$ same as before (see Figs. 2 and 9) and define $\bar{\rho}_\Xi = [\rho_1, \rho_2, \ldots, \rho_n^1, \ldots, \rho_{n+m-1}, \rho_n^2]^T$ as the *travel time vector* of the target-cycle $\bar{\Xi}$. To simplify the analysis, we skip the transient analysis of the constrained target-cycle $\bar{\Xi}$ and directly make the following assumption (see also Remark 1).

Assumption 5. The dwell time dynamics of the constrained target-cycle $\bar{\Xi}$ have a feasible and globally asymptotically stable equilibrium point.

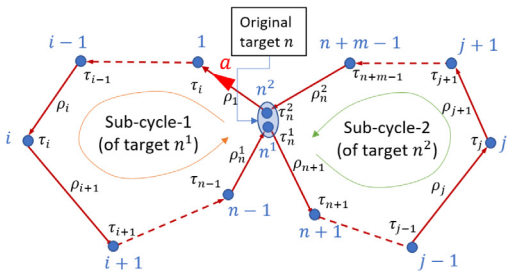


Fig. 9. A general constrained target-cycle with target n being visited twice during the cycle.

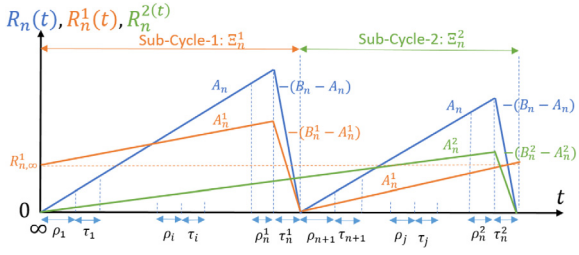


Fig. 10. Variation of the target uncertainties of the constrained target-cycle shown in Fig. 9 - after achieving steady state.

Fig. 10 shows the steady state behavior of the target uncertainties during a tour on the target-cycle $\bar{\Xi}$. The notation $\bar{\tau}_\Xi = [\tau_1, \tau_2, \ldots, \tau_n^1, \ldots, \tau_{n+m-1}, \tau_n^2]^T$ is used to represent the steady state dwell times of targets in Ξ. The following lemma generalizes Lemma 2 to evaluate $\bar{\tau}_\Xi$ for any target-cycle $\bar{\Xi}$.

Lemma 5. *Under Assumptions 2 and 5, when a single agent traverses a generic constrained target-cycle $\bar{\Xi}$ (with Ξ being the equivalent unconstrained version of $\bar{\Xi}$), the steady state dwell times $\bar{\tau}_\Xi$ are given by*

$$\bar{\tau}_\Xi = [diag(\bar{\gamma}_\Xi) - \mathbf{1}_\Xi]^{-1}\mathbf{1}_\Xi \bar{\rho}_\Xi, \tag{18}$$

where $\bar{\gamma}_\Xi \in \mathbb{R}^{|\Xi|}$ is such that if the i^{th} target of $\bar{\Xi}$ is j, then, the i^{th} element of $\bar{\gamma}_\Xi$ is $\frac{B_j}{A_j}$, and $\mathbf{1}_\Xi$ is the sub-cycle matrix and $\bar{\rho}_\Xi$ is the travel time vector of the target-cycle Ξ.

Proof. By inspection of the $R_n(t)$ profile in Fig. 10, for each target $n \in \bar{\Xi}$ and for each auxiliary target $n^j \in \mathscr{T}_n$, considering its corresponding sub-cycle Ξ_n^j's time period, we can write: $(B_n - A_n)\tau_n^j = A_n(T_n^j - \tau_n^j) \iff B_n\tau_n^j = A_nT_n^j$, where T_n^j is the total time taken to complete the sub-cycle Ξ_n^j. Now, using the sub-cycle unit vectors, we can substitute for T_n^j to get: $B_n\tau_n^j = A_n(\bar{1}_n^j)^T(\bar{\rho}_\Xi + \bar{\tau}_\Xi)$. This relationship gives $|\Xi|$ equations which we need to solve for $\bar{\tau}_\Xi \in \mathbb{R}^{|\Xi|}$. Arranging all the equations in a matrix form: $diag(\bar{\gamma}_\Xi)\bar{\tau}_\Xi = \mathbf{1}_\Xi(\bar{\rho}_\Xi + \bar{\tau}_\Xi)$ gives the result in (18). ■

Remark 1. Note that (18) is only valid under Assumption 5, i.e., if the dwell times observed in the k^{th} tour on $\bar{\Xi}$ (say $\bar{\tau}_{\Xi,k}$) converge to an equilibrium point ($\bar{\tau}_\Xi$) as $k \to \infty$. However, based on the form of (18), we can conclude that the conditions for the existence and feasibility of such an equilibrium point are $|diag(\bar{\gamma}_\Xi) - \mathbf{1}_\Xi| \neq 0$ and $[diag(\bar{\gamma}_\Xi) - \mathbf{1}_\Xi]^{-1}\mathbf{1}_\Xi\bar{\rho}_\Xi > 0$, respectively.

Remark 2. Even though we cannot apply Lemma 1 to simplify the inverse: $[diag(\bar{\gamma}_\Xi) - \mathbf{1}_\Xi]^{-1}$, using the fact that: $rank(\mathbf{1}_\Xi) =$ (number of auxiliary targets in Ξ) $\ll |\Xi|$, we can apply the Ken-Miller theorem (Miller, 1981) to efficiently compute this inverse.

Since Lemma 5 gives the dwell time vector $\bar{\tau}_\Xi$, we now can find the *total sub-cycle time* denoted by T_n^j for all targets $n^j \in \Xi$ using $T_n^j = (\bar{1}_n^j)^T(\bar{\rho}_\Xi + \bar{\tau}_\Xi)$. Moreover, we can find the *total cycle time* denoted by T_Ξ using $T_\Xi = \bar{1}_{|\Xi|}^T(\bar{\rho}_\Xi + \bar{\tau}_\Xi)$.

Lemma 6. *Under the same conditions stated in Lemma 5, the auxiliary target parameters of any $n^j \in \Xi$ (i.e., A_n^j and B_n^j) are:*

$$A_n^j = \frac{T_n^j}{T_\Xi}\frac{\tau_n^j(B_n - A_n)}{(T_\Xi - \tau_n^j)} \quad \text{and} \quad B_n^j = \frac{T_n^j(B_n - A_n)}{(T_\Xi - \tau_n^j)}. \tag{19}$$

Proof. Observing the auxiliary target uncertainty profiles $R_n^1(t)$ and $R_n^2(t)$ (of auxiliary targets n^1 and n^2, respectively) illustrated in Fig. 10 for the target-cycle shown in Fig. 9, note that the shape of these profiles should satisfy the equivalence criteria that we previously established. Therefore, for any generic target-cycle $\bar{\Xi}$, the first equivalence criterion is guaranteed by:

$$B_n^j\tau_n^j = A_n^jT_n^j, \quad \forall n^j \in \mathscr{T}_n, \ \forall n \in \bar{\Xi}. \tag{20}$$

Using (2), we can write the contribution from a target $n \in \bar{\Xi}$ to the main objective J_T during a total cycle time period (of length T_Ξ, at steady state) as $\frac{1}{T}\int_{T_\Xi} R_n(t)dt$. Therefore, to enforce the third equivalence criterion, we need: $\frac{1}{T}\int_{T_\Xi} R_n(t)dt = \frac{1}{T}\int_{T_\Xi}\sum_{n^j\in\mathscr{T}_n} R_n^j(t)dt$. However, since we can decompose T_Ξ into sub-cycle time periods, we can write: $\frac{1}{T}\int_{T_\Xi} R_n(t)dt = \frac{1}{T}\sum_{n^j\in\mathscr{T}_n}\int_{T_n^j} R_n(t)dt$. We use these two relationships to create a system of equations: $\int_{T_\Xi} R_n^j(t)dt = \int_{T_n^j} R_n(t)dt$, $\forall n^j \in \mathscr{T}_n$, $\forall n \in \bar{\Xi}$. As uncertainty profiles are piece-wise linear, we can evaluate these integrals and simplify this system of equations as:

$$T_\Xi(B_n^j - A_n^j) = T_n^j(B_n - A_n), \quad \forall n^j \in \mathscr{T}_n, \ \forall n \in \bar{\Xi}. \tag{21}$$

Finally, we can solve (20) and (21) to obtain the auxiliary target parameters: $\{(A_n^j, B_n^j) : \forall n^j \in \mathscr{T}_n, \ \forall n \in \bar{\Xi}\}$ as in (19). ■

Using the auxiliary target parameters given by Lemma 6, we now lump all the respective parameters of targets in Ξ into vectors as $\bar{A}_\Xi$ and $\bar{B}_\Xi$. For example, for the target-cycle shown in Fig. 9, $\bar{A}_\Xi = [A_1, A_2, \ldots, A_n^1, \ldots, A_{n+m-1}, A_n^2]^T$.

Theorem 3. *Under Assumptions 2 and 5, when a single agent traverses a generic constrained target-cycle $\bar{\Xi}$ (with Ξ being the equivalent unconstrained version of $\bar{\Xi}$), the steady state mean cycle uncertainty $J_{ss}(\bar{\Xi})$ (defined in (8)) is*

$$J_{ss}(\bar{\Xi}) = \frac{1}{2}(\bar{B}_\Xi - \bar{A}_\Xi)^T\bar{\tau}_\Xi, \tag{22}$$

where $\bar{\tau}_\Xi$ is given by Lemma 5 and auxiliary target parameters included in the vectors $\bar{A}_\Xi$ and $\bar{B}_\Xi$ are given by Lemma 6.

Proof. Note that Ξ is an unconstrained target-cycle. Therefore, we can directly use Theorem 1 to write $J_{ss}(\Xi) = \frac{1}{2}(\bar{B}_\Xi - \bar{A}_\Xi)^T\bar{\tau}_\Xi$, where $\bar{\tau}_\Xi$ is given by Lemma 5 and unknown parameters in $\bar{A}_\Xi$ and $\bar{B}_\Xi$ are given by Lemma 6. Finally, according to the equivalence criterion 3: $J_{ss}(\bar{\Xi}) = J_{ss}(\Xi)$. ■

4.2. Greedy target-cycle construction

Let $\mathscr{D}$ denote the set of all possible target-cycles on $\mathscr{G}$. Compared to $\mathscr{C}$ in (15), $\mathscr{D} \supseteq \mathscr{C}$ as $\mathscr{D}$ now also includes all the constrained target-cycles. Clearly, $|\mathscr{D}| = \infty$ and thus, exhaustive search methods (exploiting Theorem 3) cannot be used to determine the best performing target-cycle in $\mathscr{D}$:

$$\bar{\Xi}^* = \arg\min_{\bar{\Xi}\in\mathscr{D}} J_{ss}(\bar{\Xi}). \tag{23}$$

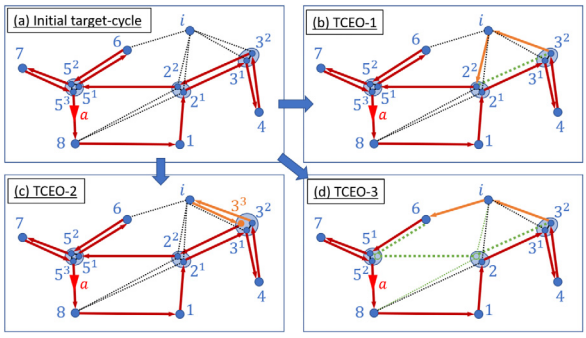


Fig. 11. Target-cycle expansion operations (TCEOs).

Hence, we seek to efficiently construct a sub-optimal target-cycle $\bar{\Xi}^{\#} \in \mathscr{D}$ as a candidate for $\bar{\Xi}^{*}$ in (23) using a greedy iterative target-cycle expansion process identical to Alg. 2.

Notice that having the capability to make repeated visits to the targets during a tour on a target-cycle (in other words, the capability to add auxiliary targets) provides flexibility in ways we can expand a given target-cycle. Therefore, in this paradigm, we are not constrained to use only the basic target-cycle expansion operation (labeled TCEO-1) used before – shown in Fig. 4 and copied in Fig. 11(b). Motivated by this flexibility, we propose two additional TCEOs labeled TCEO-2 and TCEO-3 shown Fig. 11(c) and (d), respectively. In particular, TCEO-2 adds a neglected target to the current target-cycle by creating a new auxiliary target. For example, compare Fig. 11(a) and (c) to see that the neglected target i is only included in the expanded target-cycle via the newly added auxiliary target 3^3. In contrast, TCEO-3 adds a neglected target to the current target-cycle by eliminating one or more existing auxiliary targets. For example, compare Fig. 11(a) and (d) to see that the neglected target i is only included in the expanded target-cycle via the removal of the auxiliary targets 2^2 and 5^1.

Regardless of the type of the TCEO, note that we can use Theorems 3 and 2 to determine the corresponding marginal gain. This enables us to use an identical target-cycle construction algorithm to the one shown in Alg. 2 for the purpose of constructing a sub-optimal target-cycle $\bar{\Xi}^{\#} \in \mathscr{D}$ as a candidate for $\bar{\Xi}^{*}$ in (23). However, note that in each greedy target-cycle expansion iteration (i.e., in Step 3 of Alg. 2), we now need to determine the best feasible target-cycle expansion considering all three types of TCEOs not limiting to TCEO-1. These additional two greedy search space dimensions introduced (due to TCEO-2 and TCEO-3) resolve the issue of 'premature halting' of the target-cycle expansion process, because there is always a feasible target-cycle expansion that belongs to the type TCEO-2 whenever there are neglected targets (i.e., when $|\mathscr{V}\backslash\bar{\Xi}| > 0$) and the network is connected.

TSP-inspired target-cycle refinements. If the sub-optimal target-cycle $\bar{\Xi}^{\#}$ given by the greedy target-cycle construction algorithm is an unconstrained one, we can directly apply the 2-Opt and 3-Opt techniques adopted from the TSP literature (Blazinskas & Misevicius, 2011) to refine the solution $\bar{\Xi}^{\#}$ further. However, if $\bar{\Xi}^{\#}$ is a constrained target-cycle, application of such 2-Opt or 3-Opt techniques requires a few additional steps that we omit discussing here, but can be found in Welikala and Cassandras (2019). In the sequel, we denote the refined version of $\bar{\Xi}^{\#}$ as $\bar{\Xi}^{R}$.

4.3. Generating an initial TCP: $\Theta^{(0)}$

Recall that our final goal is to transform $\bar{\Xi}^{R}$ into a set of TCP parameters and then to use them as the initial condition $\Theta^{(0)}$ in the gradient descent process (7). Due to the single-agent assumption (i.e., Assumption 1(i)), we can write $\mathscr{A} = \{a\}$ and $\Theta^{(0)} = \Theta^{a(0)}$. Note that even though $\bar{\Xi}^{R}$ might be a constrained target-cycle, we still can use Alg. 3 to get the corresponding TCP $\Theta^{a(0)}$, but under few minor conditions (provided in Welikala & Cassandras, 2019). Fig. 12 shows an example constrained target-cycle and its corresponding TCP $\Theta^{a(0)}$ given by Alg. 3.

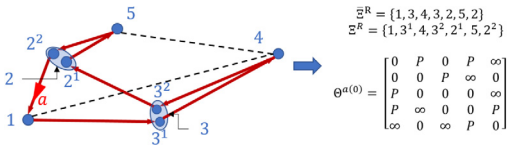


Fig. 12. The generated threshold matrix $\Theta^{a(0)}$ for the refined sub-optimal target-cycle $\bar{\Xi}^{R}$ shown (left).

5. Multi-agent PMN solution

In the previous two sections, we focused on the single-agent PMN problem and developed techniques to (i) identify a favorable agent trajectory in a given network and (ii) transform the identified trajectory into a TCP $\Theta^{(0)}$ for the subsequent use in a gradient process (7). Now, in order to conveniently generalize these single-agent techniques to multi-agent PMNs, as outlined in Alg. 1, we will partition the network $\mathscr{G}$ into N sub-graphs (recall $N = |\mathscr{A}|$). This 'divide and conquer' approach allows us to use the developed single-agent techniques (that respectively correspond to Steps 2 and 5 of Alg. 1) independently in each of the sub-graphs. Therefore, this section presents the proposed graph partitioning, refining and agent assigning processes that respectively correspond to Steps 1, 3 and 4 of Alg. 1 (i.e., of the overall PMN solution). However, for these processes, we largely adapt known techniques from Ahuja, Magnanti, and Orlin (1993), Ng, Jordan, and Weiss (2001), Shi and Malik (2000) and von Luxburg (2007), and thus we omit some technical details in this section (which can be found in Welikala & Cassandras, 2019) so as to emphasize our own contributions to achieve this adaptation. Before getting into details, we emphasize again that each of these processes is conducted only to arrive at favorable initial TCP efficiently, and hence the subsequent use of gradient descent (7) may compensate for any sub-optimalities in the used processes.

5.1. Graph partitioning using spectral clustering

To partition the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, we use the *spectral clustering* (von Luxburg, 2007) technique – which is a commonly used global graph partitioning method that also has the advantages of: (i) simple implementation, (ii) efficient evaluation and (iii) better results compared to traditional techniques such as the k-means algorithm (von Luxburg, 2007). In spectral clustering, the graph partitions of $\mathscr{G}$ are derived based on a set of inter-target *similarity* values $\{s_{ij} : i, j \in \mathscr{V}\}$ so that the similarity value between two targets is high if they belong to the same partition and low otherwise.

Remark 3. In a typical data-point clustering application, the graph representation (also called the "similarity graph") arises from the known similarity values between the data-points. However, this is not the case for PMN problems where the physical graph $\mathscr{G}$ (which is different from the similarity graph) is known, while the similarity values between its targets are unknown.

Deriving similarity values. We exploit the knowledge of the target topology $\mathscr{G}$ and target parameters to derive appropriate similarity values. Typically, a similarity value $s_{ij} \geq 0$ is obtained based on a *disparity* value $d(i,j)$ as

$$s_{ij} = \exp\left(-\frac{|d(i,j)|^2}{2\sigma^2}\right), \quad i,j \in \mathscr{V}, \tag{24}$$

where $d : \mathscr{V} \times \mathscr{V} \rightarrow \mathbb{R}$ and σ^2 is a user defined scaling parameter that controls how rapidly the similarity s_{ij} falls off with the disparity between i and j (i.e., with $d(i,j)$) (Ng et al., 2001). This function (24) is known as the *Gaussian similarity function*. Note also that the disparity and similarity values are inversely related. We next focus on defining an appropriate disparity metric for the PMN problems.

For the considered PMN problem setup, neither of using $d(i,j)$ as the *physical distance* (i.e., $\|X_i - X_j\|$) nor the *shortest distance* between the targets i and j provides a good characterization to the underlying persistent monitoring aspects of the problem because they disregard target parameters and agent behaviors when monitoring targets.

Therefore, we propose a novel disparity metric named *minimum mean covering cycle uncertainty* (CCU):

$$d(i,j) = d_{CC}(i,j) \triangleq \min_{\bar{\Xi}: i,j \in \bar{\Xi}} J_{ss}(\bar{\Xi}). \tag{25}$$

The arg min of the above problem is named the *optimal covering cycle* (OCC) and we denote it as $\bar{\Xi}^*_{ij}$. In other words, the OCC $\bar{\Xi}^*_{ij}$ is the best way to cover targets i and j in a single target-cycle so that the corresponding steady state mean cycle uncertainty is minimized. Therefore, if the CCU value is higher for a certain target pair, it implies that it is difficult to cover those two targets in a single target-cycle. Hence, it is clear that this disparity metric $d_{CC}(i,j)$ in (25) provides a good characterization to the underlying persistent monitoring aspects of the PMN problems. As an example, if all the trajectory segments in $\mathscr{E}$ follow the triangle inequality in terms of respective travel times, then, for any $(i,j) \in \mathscr{E}$, the corresponding OCC is $\bar{\Xi}^*_{ij} = \{i,j\}$ and CCU is $J_{ss}(\{i,j\})$.

Motivated by the above example, to estimate the disparity metric values: $d_{CC}(i,j)$, $\forall i,j \in \mathscr{V}$ we propose a modified version of the Dijkstra's algorithm (Ahuja et al., 1993) coupled with the cycle expanding and refining techniques discussed in Section 4 (details are omitted for brevity, but can be found in Welikala & Cassandras, 2019). The resulting disparity values are then transformed using (24) to obtain the respective similarity values s_{ij} $\forall i,j \in \mathscr{V}$.

Spectral clustering algorithm. Finally, based on these similarity values, the normalized spectral clustering technique proposed in Shi and Malik (2000) is applied to derive the set of target partitions of $\mathscr{V}$, i.e., $\{\mathscr{V}_a : a \in \mathscr{A}\}$ and the respective sub-graphs $\{\mathscr{G}_a : a \in \mathscr{A}\}$ where each $\mathscr{G}_a = (\mathscr{V}_a, \mathscr{E}_a)$ and $\mathscr{E}_a \subseteq \mathscr{E}$ is the set of intra-cluster edges (details are provided in Welikala & Cassandras, 2019).

5.2. Refining the graph partitions

Once the sub-graphs are formed (i.e., Step 1 of Alg. 1), we execute the target-cycle construction procedure (i.e., Step 2 of Alg. 1, presented in Section 4) on each sub-graph. The resulting target-cycle on a sub-graph $\mathscr{G}_a$ is denoted as $\bar{\Xi}_a$ and is assumed to be assigned to an arbitrary agent $a \in \mathscr{A}$. Note, however, that in Section 5.3, we will explicitly assign target-cycles to the agents.

An agent $a \in \mathscr{A}$ can remove a target $i \in \bar{\Xi}_a$ from its target-cycle $\bar{\Xi}_a$ by reconstructing a new target-cycle over its sub-graph $\mathscr{G}_a$ while ignoring target i. We call such a process a target-cycle *contraction*. In contrast, an agent $b \in \mathscr{A}$ can expand its target-cycle $\bar{\Xi}_b$ to include an external target $i \notin \bar{\Xi}_b$ by simply carrying out the best possible target-cycle *expansion* out of the three TCEOs shown in Fig. 11. Using such contraction and expansion operations, two agents a, $b \in \mathscr{A}$ can *trade* a target $i \in \bar{\Xi}_a$ between each other, if the marginal gain $\Delta J^{ab,i}_{ss} \triangleq (J_{ss}(\bar{\Xi}_a) + J_{ss}(\bar{\Xi}_b)) - (J_{ss}(\bar{\Xi}'_a) + J_{ss}(\bar{\Xi}'_b)) > 0$, where $\bar{\Xi}'_a$ and $\bar{\Xi}'_b$ are the contracted and expanded target-cycles, respectively. Upon such a trade, the agents can update their sub-graphs $\mathscr{G}_a$ and $\mathscr{G}_b$ appropriately.

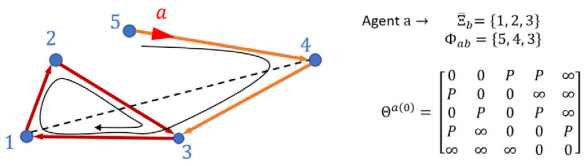


Fig. 13. The generated initial TCP $\Theta^{a(0)}$ when the agent a is initially at target 5 and have been assigned to the target-cycle $\bar{\Xi}_b = \{3,1,2\}$ with the fastest path being $\Phi_{ab} = \{5,4,3\}$.

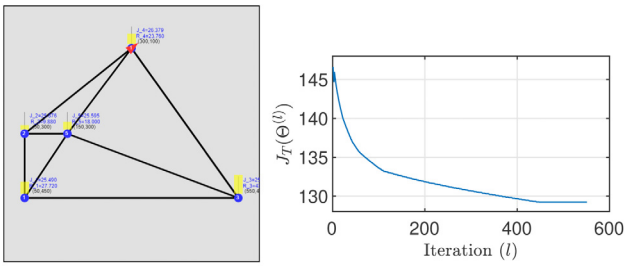
We call a set of sub-graphs "balanced" if there is no $a, b \in \mathscr{A}$ and $i \in \mathscr{V}_a$ such that $\Delta J^{ab,i}_{ss} > 0$. The spectral clustering method used often provides a balanced set of sub-graphs. However, when this is not the case, we propose a distributed greedy algorithm (details are provided in Welikala & Cassandras, 2019) for the agents to use so as to balance the sub-graphs by systematically executing trades with positive marginal gains. The convergence of such a greedy algorithm can be ascertained by observing the fact that each greedy step (i.e., each 'trade') decreases the metric: $\sum_{a \in \mathscr{A}} J_{ss}(\bar{\Xi}_a)$, which is lower bounded by 0. With that, we conclude the discussion about Step 3 of Alg. 1.

5.3. Assigning agents to the target-cycles

So far, we have identified a set of target-cycles $\{\bar{\Xi}_b : b \in \mathscr{B}\}$ on the corresponding set of balanced sub-graphs of $\mathscr{G}$, where $\mathscr{B}$ is the set of target-cycle indexes (identical to the set $\mathscr{A}$). We now explicitly assign these target-cycles $\{\bar{\Xi}_b : b \in \mathscr{B}\}$ to the agents based on initial agent locations $\{s_a(0) : a \in \mathscr{A}\}$. First, let us define the assignment cost between an agent $a \in \mathscr{A}$ and a target-cycle $\bar{\Xi}_b$, $b \in \mathscr{B}$ as h_{ab} where h_{ab} represents the total travel time on the fastest available path to reach any one of the targets in $\bar{\Xi}_b$ starting from $s_a(0)$. We use Dijkstra's shortest path algorithm (Ahuja et al., 1993) to compute all these assignment weights. Subsequently, the assignment problem (between a's and b's) is solved using the shortest augmenting path algorithm (Ahuja et al., 1993).

Generating an initial TCP: $\Theta^{(0)}$. Let us assume agent $a \in \mathscr{A}$ is optimally assigned to the target-cycle $\bar{\Xi}_b$ and the corresponding fastest path from $s_a(0)$ to reach $\bar{\Xi}_b$ is $\Phi_{ab} = \{i_1, i_2, \ldots, i_n\} \subset \mathscr{V}$. Note that $i_n \in \bar{\Xi}_b$ and $X_{i_1} = s_a(0)$. Next, let us define $\Phi'_{ab} = \Phi_{ab} \backslash \{i_n\}$. We now use Alg. 3 with $\bar{\Xi}_b$ to get a corresponding TCP for agent a as Θ^a. Note that Alg. 3 only assigns the set of rows: $\{j : j \in \bar{\Xi}_b\}$ in Θ^a as it is sufficient to keep the agent on the target-cycle $\bar{\Xi}_b$ (this corresponds to rows 1–3 in the example TCP Θ^a shown in Fig. 13). Therefore, to make sure that agent a follows the path Φ_{ab}, we assign the set of rows: $\{j : j \in \Phi'_{ab}\}$ in Θ^a as follows (this corresponds to rows 4–5 in the example TCP Θ^a shown in Fig. 13). If j and k are two consecutive entries in Φ_{ab}, in the j^{th} row of Θ^a we set: $\theta^a_{jj} = 0$, $\theta^a_{jk} = 0$ and any other entry θ^a_{jl} is set to P or ∞ depending on whether $(j,l) \in \mathscr{E}$. Finally, we set $\Theta^{a(0)} = \Theta^a$.

This concludes the discussion about Steps 4 and 5 of Alg. 1, and thus, we have now covered all the steps involved in Alg. 1 - which is the proposed PMN solution in this paper.



(a) Config. at $t = T$. (b) Cost vs iterations plot.

Fig. 14. Single-agent simulation example 1 (SASE1): Started with a random $\Theta^{(0)}$, converged to a TCP with $J_T = 129.2$.

Remark 4. In some persistent monitoring applications, agents have limited energy and need to be occasionally recharged. Thus, it is crucial to plan for agent recharging periods (also known as incorporating "endurance constraints"). One approach to incorporate such endurance constraints is by appropriately selecting the planning horizon length T so that all the agents can endure such a period. Another approach is to consider the available charging stations as targets with relatively slower uncertainty growth rates so as to ensure occasional agent visits there. Details are beyond this paper's scope and part of ongoing research.

6. Simulation results

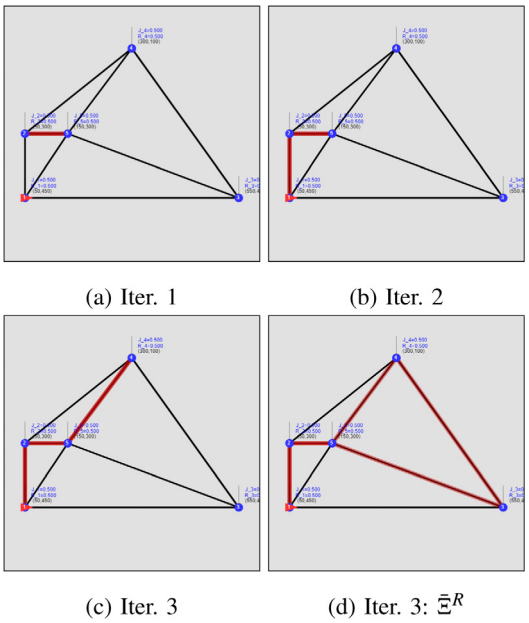
In this section, we provide several numerical examples to show how the greedy initialization process we have developed can benefit the performance of the TCP used in solving PMN problems. As a starting point, consider the single-agent PMN problem configuration (labeled SASE1) shown in Fig. 14(a). In this figure (and similar figures used in the sequel), blue circles represent the targets, while black lines represent available trajectory segments that agents can take to travel between targets. Red triangles and the yellow vertical bars respectively indicate the agent locations and the target uncertainty levels. Since both of those quantities are time-varying (i.e., $s_a(t)$ and $R_i(t)$), we only indicate their state at the terminal time $t = T$ in a simulation where the best TCP found so far is used. In all numerical examples, the PMN problem parameters have been chosen as follows. The target parameters are: $A_i = 1,\ B_i = 10,\ R_i(0) = 0.5,\ \forall i \in \mathscr{V}$ and the target locations (i.e., X_i) are specified in each problem configuration figure. In all the examples, all the targets have been placed inside a 600×600 mission space. The time horizon is taken as $T = 500$. Each agent is assumed to have first-order dynamics (following from Zhou et al., 2019) with a maximum speed of 50 units per second. The initial locations of the agents are chosen such that they are uniformly distributed among the targets at $t = 0$ (i.e., $s_a(0) = X_i$ with $i = 1 + (a-1) * \text{round}(M/N)$). In cases where the initial TCP $\Theta^{(0)}$ is randomly generated, each finite element in $\Theta^{(0)}$ is chosen from the uniform distribution unif(0, 10).

The proposed PMN solution in this paper (i.e., Alg. 1) including the method proposed in Zhou et al. (2019) have been implemented in a JavaScript-based interactive simulation platform available at http://www.bu.edu/codes/simulations/shiran27/PersistentMonitoring/. Readers are invited to reproduce the reported results and also to try new problem configurations using this simulator.

Fig. 14(b) shows the evolution of $J_T(\Theta^{(l)})$ in the SASE1 when the TCP $\Theta^{(l)}$ was updated according to (7) starting from a randomly selected initial TCP $\Theta^{(0)}$ as proposed in Zhou et al. (2019).

We next apply the PMN solution proposed in this paper to the SASE1. First, a high-performing target-cycle was constructed using the proposed greedy algorithm (Alg. 2).Fig. 15(a)→(d) show the intermediate target-cycles observed (as red traces) during this process. The resulting target-cycle in Fig. 15(d) is $\bar{\Xi}^R = \{2, 1, 2, 5, 3, 4, 5\}$ with $J_{ss}(\bar{\Xi}^R) = 121.1$. Then, $\bar{\Xi}^R$ was transformed into a TCP $\Theta^{(0)}$ using Alg. 3 to initialize the gradient descent (7). Fig. 16(b) shows that the obtained TCP $\Theta^{(0)}$ results in a J_T value of 114.9 which is not further improved via the gradient descent (7), i.e., $\Theta^{(0)}$ may be directly locally optimal. To ensure this, after 100 iterations of (7) (at $l = 100$), $\Theta^{(l)}$ was randomly perturbed and it was observed that $\Theta^{(l)}$ converges back to the same initial TCP (with $J_T = 114.9$). Regardless, we highlight that this solution is 11.1% better than the solution given by Zhou et al. (2019) (shown in Fig. 14).

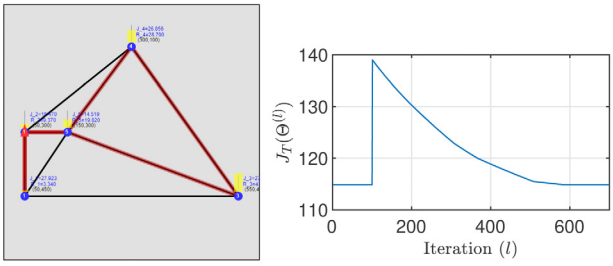
The second simulation example we consider is the single-agent PMN problem configuration shown in Fig. 17(a) (labeled SASE2). In particular, we use SASE2 to highlight the importance of gradient steps (7). As shown in Fig. 17(b), when the initial TCP $\Theta^{(0)}$ was selected randomly, (7) converged to a TCP with $J_T = 651.3$. In contrast, when $\Theta^{(0)}$ was derived using the technique proposed in this paper, it directly yields $J_T = 607.9$. Next, when (7) is used to iteratively update the TCP, unlike in SASE1, we



(a) Iter. 1 (b) Iter. 2

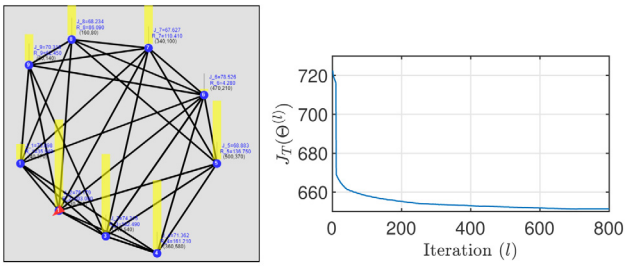
(c) Iter. 3 (d) Iter. 3: $\bar{\Xi}^R$

Fig. 15. Greedy target-cycle construction for the SASE1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



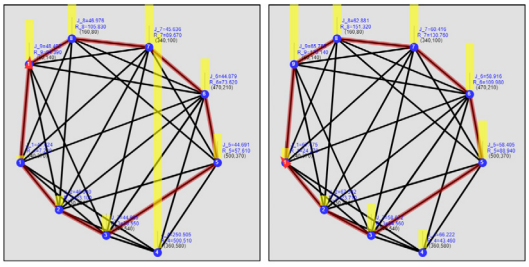
(a) Config. at $t = T$. (b) Cost vs iterations plot.

Fig. 16. SASE1: The TCP $\Theta^{(0)}$ given by the target-cycle $\bar{\Xi}^R$ (the red trace in (a)) shows local optimality. At $l = 100$, $\Theta^{(l)}$ is randomly perturbed. Yet, converges back to the initial TCP. Cost $J_T = 114.9$ (Improvement $= +14.3$ compared to Fig. 14). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

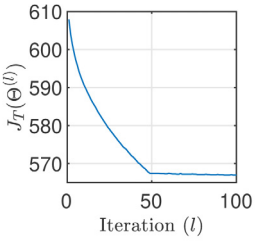


(a) Config. at $t = T$. (b) Cost vs iterations plot.

Fig. 17. Single-agent simulation example 2 (SASE2): Started with a random $\Theta^{(0)}$ and converged to a TCP with $J_T = 651.3$.



(a) $l = 0$, $t = T$. (b) $l = 100$, $t = T$.

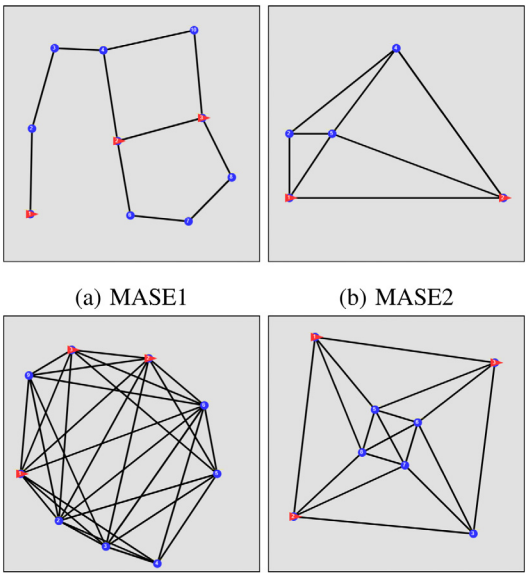


(c) Cost vs iterations.

Fig. 18. SASE2: The derived initial TCP $\Theta^{(0)}$ has a cost $J_T = 607.9$ and is further improved by (7) to reach TCP with a final cost $J_T = 567.0$ (Improvement $= +84.3$ compared to Fig. 17). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

observed a further improvement in J_T, finally reaching $J_T = 567.0$ (see Fig. 18(b) and (c)). The main difference between the solutions in Fig. 18(a) and (b) is that in the former the agent avoids visiting target 4 and strictly follows the target-cycle shown in red color, whereas in the latter gradient descent steps have updated the TCP such that the agent trajectory now includes target 4. Compared to Zhou et al. (2019), the percentage improvement achieved from deploying the proposed PMN solution is 12.9%.

We now consider four multi-agent PMN problem configurations shown in Fig. 19(a)–(d) as our multi-agent simulation examples (MASEs). Note that in MASE2, only two agents were deployed, whereas in all the rest three agents were deployed. As shown in Fig. 20, when the initial TCP $\Theta^{(0)}$ was chosen randomly, the gradient steps (7) have converged to TCPs with J_T values 270.2, 91.7, 274.0, and 201.3 in respective MASE. The sub-graphs obtained from the proposed graph partitioning technique (i.e., Step 1 of Alg. 1) for each of the MASEs are shown in Fig. 21. The constructed target-cycles in sub-graphs and the process of sub-graph refinement (i.e., Steps 2 and 3 of Alg. 1) are demonstrated in Fig. 22(a)–(d) with regard to the MASE1. Sub-figures in Fig. 23 show the determined respective graph partitions and target-cycles. It was observed that the initial TCPs given by Alg. 1 are directly locally optimal (similar to what we



(a) MASE1 (b) MASE2

(c) MASE3 (d) MASE4

Fig. 19. Multi-agent simulation examples (MASEs) at $t = 0$.

saw in SASE1). However, these TCPs performed better than the optimal TCPs obtained with randomly initialized $\Theta^{(0)}$ (shown in Fig. 20). In particular, the percentage improvements achieved are: 66.3%, 61.7% 78.2%, and 70.3%, respectively. All the discussed simulation results so far have been summarized in Table 1.

Next, we consider eight randomly generated MASEs (with $N = 3$, $M = 15$, see Welikala & Cassandras, 2019 for details) (Dall & Christensen, 2002). When the proposed PMN solution (i.e., Alg. 1) was deployed, across these eight MASEs, the average percentage improvement achieved was 69.1%. In fact, on an Intel® Core™ i7-7800 CPU 3.20 GHz Processor with a 32 GB RAM, the average execution time taken for the proposed Alg. 1 to generate the TCP $\Theta^{(0)}$ was 13.7 s and all such generated TCPs were immediately locally optimal (similar to what we saw in SASE1, MASE1-MASE4). In contrast, when the TCPs $\Theta^{(0)}$ were randomly generated, the average execution time observed for the convergence of the gradient steps (7) was 245.8 s. Therefore, the execution time taken for the proposed off-line greedy initialization process is much smaller and, at the same time, highly effective.

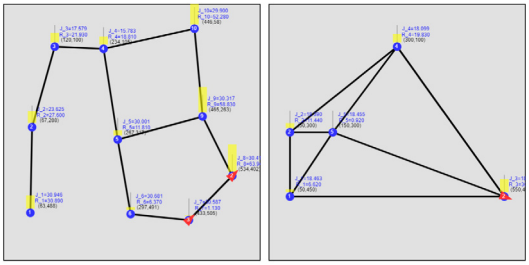
Comparison with a Brute-Force approach. Finally, to get a sense of how close the proposed PMN solution is to the global optimal solution, we consider three simple SASEs shown in Fig. 24 and compare the observed performance levels of the proposed solution to those of a brute-force (exhaustive search) solution.

In particular, this brute-force solution uses Assumptions 1-(i) and 2 to reduce the search space size. As a result, it only requires to search (exhaustively) for the optimal visiting target sequence for a single agent in the given network. However, the search space size still exponentially increases with the number of target visits required (which typically is around 100 target visits for these examples). To address this challenge, when generating candidate visiting target sequences, we use a branch and bound approach and keep only a finite number of candidate solutions at any iteration. Since this cannot guarantee finding a globally optimal solution, we limit ourselves to simple cases (such as the ones shown in Fig. 24).

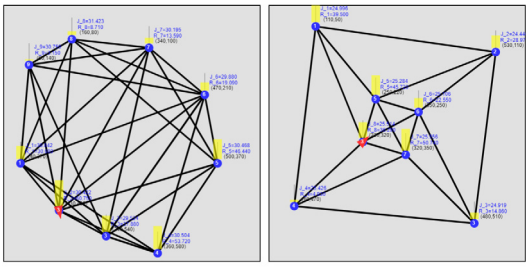
Based on the observed results reported in Fig. 24, it is clear that the proposed solution scheme in this paper achieves virtually the same performance levels while only requiring a fraction of the execution time to determine the brute-force (globally optimal) solutions.

Table 1
A summary of obtained simulation results (see also (Welikala & Cassandras, 2019)).

Cost of the optimal TCP Θ^* (found using (7)): $J_T(\Theta^*)$	Single-agent simulation examples			Multi-agent simulation examples			
	1	2	3	1	2	3	4
With randomly generated initial TCP $\Theta^{(0)}$	129.2	651.3	497.9	270.2	91.7	274.0	201.3
With initial TCP $\Theta^{(0)}$ given by the proposed Alg. 1	114.6	567.0	449.5	90.9	35.1	59.5	59.8
Percentage improvement (%)	11.1	12.9	9.7	66.3	61.7	78.2	70.3

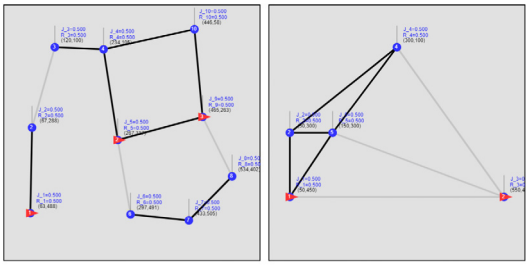


(a) MASE1 $J_T = 270.2$ (b) MASE2 $J_T = 91.7$

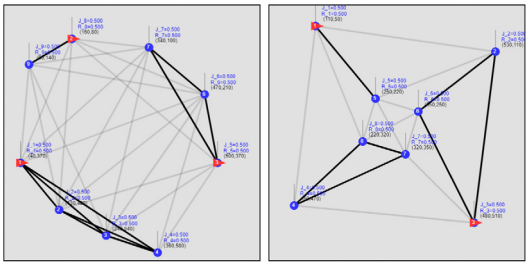


(c) MASE3 $J_T = 274.0$ (d) MASE4 $J_T = 201.3$

Fig. 20. Cost J_T achieved in each MASE upon convergence when started with a random $\Theta^{(0)}$ (Config. at $t = T$ is shown).

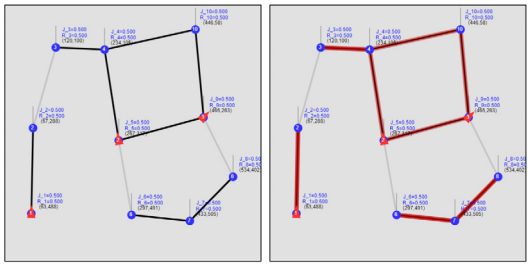


(a) MASE1 (b) MASE2

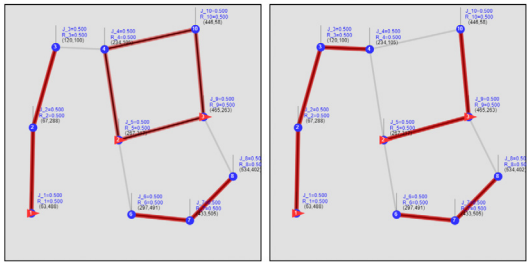


(c) MASE3 (d) MASE4

Fig. 21. Clustering results obtained for the considered MASEs.

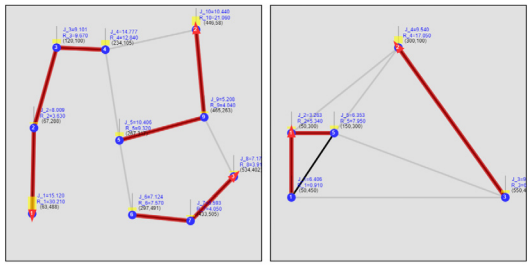


(a) Clusters (b) Cycles

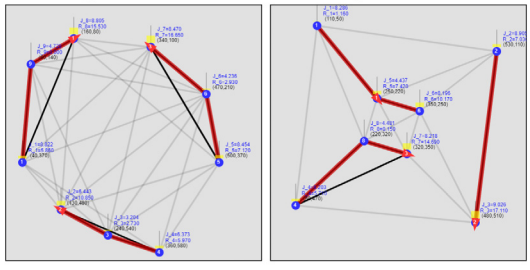


(c) Trade 1 (d) Trade 2

Fig. 22. For the MASE1: (a) initial sub-graphs, (b) initial target-cycles, (c)–(d) two 'trading' steps and (d) the final sub-graph/target-cycle arrangement.

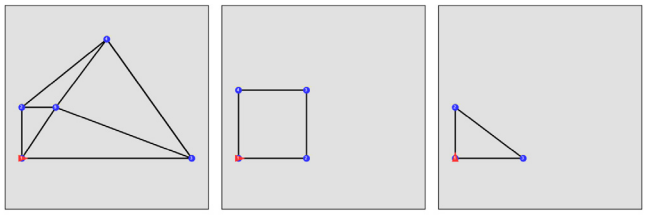


(a) MASE1 $J_T = 90.9$ Impr.$= +179.3$ (b) MASE2 $J_T = 35.1$ Impr.$= +56.6$



(c) MASE3 $J_T = 59.5$ Impr.$= +214.5$ (d) MASE4 $J_T = 59.8$ Impr.$= +141.5$

Fig. 23. Cost J_T and improvement achieved in each MASE compared to Fig. 20. Each MASE started (7) with the TCP $\Theta^{(0)}$ given by Alg. 1 and found that $\Theta^{(0)}$ is directly locally optimal (Config. at $t = T$ is shown).



(a) SASE1
Proposed:
$J_T = 114.9$
Brute Force:
$J_T = 114.1$

(b) SASE4
Proposed:
$J_T = 47.01$
Brute Force:
$J_T = 47.10$

(c) SASE5
Proposed:
$J_T = 22.30$
Brute Force:
$J_T = 22.33$

Fig. 24. Simple SASEs (at $t = 0$) used for the performance comparison between the proposed solution and the brute force solution. Observed performance levels are given under each caption. Average percentage performance difference: 0.320%. Average percentage execution time saved: 98.02%.

7. Conclusion

We have considered the optimal multi-agent persistent monitoring problem on a set of targets interconnected according to a fixed graph topology. We have adopted a class of distributed threshold-based parametric controllers where IPA can be used to determine optimal threshold parameters in an on-line manner using gradient descent. Due to the non-convex nature of the problem, optimal thresholds given by gradient descent highly depend on the used initial thresholds. To address this issue, the asymptotic behavior of the persistent monitoring system was studied, which leads to a computationally efficient and effective threshold initialization scheme. Future work is directed at extending the proposed solution to PMN problems with variable travel times given by higher-order agent dynamic models.

References

Ahuja, Ravindra K., Magnanti, Thomas L., & Orlin, James B. (1993). *Network flows : Theory, algorithms, and applications*. Prentice Hall.

Aksaray, Derya, Leahy, Kevin, & Belta, Calin (2015). Distributed multi-agent persistent surveillance under temporal logic constraints. *IFAC-PapersOnLine*, *48*(22), 174–179.

Alamdari, Soroush, Fata, Elaheh, & Smith, Stephen L. (2014). Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations. *International Journal Robotics Research*, *33*(1), 138–154. http://dx.doi.org/10.1177/0278364913504011.

Bektas, Tolga (2006). The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega*, *34*(3), 209–219.

Bertsekas, D. P. (2016). *Nonlinear programming*. Athena Scientific.

Blazinskas, Andrius, & Misevicius, Alfonsas (2011). *Combining 2-Opt, 3-Opt and 4-Opt with K-swap-kick perturbations for the travelling salesman problem*: *Technical report*, URL: https://api.semanticscholar.org/CorpusID:15324387.

Bof, Nicoletta, Carli, Ruggero, & Schenato, Luca (2018). Lyapunov theory for discrete time systems. arXiv e-prints, 1809.05289. URL: http://arxiv.org/abs/1809.05289, arXiv:1809.05289.

Caprari, Gilles, Breitenmoser, Andreas, Fischer, Wolfgang, Hürzeler, Christoph, Tâche, Fabien, Siegwart, Roland, et al. (2010). Highly compact robots for inspection of power plants. In *Intl. conf. on applied robotics for the power industry*.

Cassandras, Christos G., & Lafortune, Stephane (2010). *Introduction to discrete event systems* (2nd ed.). Springer Publishing Company, Inc..

Cassandras, Christos G., Wardi, Yorai, Panayiotou, Christos G., & Yao, Chen (2010). Perturbation analysis and optimization of stochastic hybrid systems. *European Journal of Control*, *16*(6), 642–661.

Dall, Jesper, & Christensen, Michael (2002). Random geometric graphs. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, *66*(1).

Fan, Fei, Wu, Gongping, Wang, Man, Cao, Qi, & Yang, Song (2018). Multi-robot cyber physical system for sensing environmental variables of transmission line. *Sensors (Switzerland)*, *18*(9).

Hari, S. K. K., Rathinam, S., Darbha, S., Kalyanam, K., Manyam, S. G., & Casbeer, D. (2019). The generalized persistent monitoring problem. In *Proc. of American control conf.* (pp. 2783–2788).

Hari, Sai Krishna Kanth, Rathinam, Sivakumar, Darbha, Swaroop, Kalyanam, Krishna, Manyam, Satyanarayana Gupta, & Casbeer, David (2021). Optimal UAV route planning for persistent monitoring missions. *IEEE Transactions on Robotics*, *37*(2), 550–566. http://dx.doi.org/10.1109/TRO.2020.3032171.

Held, Michael, & Karp, Richard M. (1971). The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, *1*(1), 6–25. http://dx.doi.org/10.1007/BF01584070.

Huynh, Vu Anh, Enright, John J., & Frazzoli, Emilio (2010). Persistent patrol with limited-range on-board sensors. In *Proc. of 49th IEEE conf. on decision and control* (pp. 7661–7668).

Khazaeni, Yasaman, & Cassandras, Christos G. (2018). Event-driven cooperative receding horizon control for multi-agent systems in uncertain environments. *IEEE Transactions on Control of Network Systems*, *5*(1), 409–422.

Lan, Xiaodong, & Schwager, Mac (2013). Planning periodic persistent monitoring trajectories for sensing robots in Gaussian random fields. In *Proc. of IEEE intl. conf. on robotics and automation* (pp. 2415–2420). http://dx.doi.org/10.1109/ICRA.2013.6630905.

Liaqat, A., Hutabarat, W., Tiwari, D., Tinkler, L., Harra, D., Morgan, B., et al. (2019). Autonomous mobile robots in manufacturing: Highway code development, simulation and testing. *International Journal of Advanced Manufacturing Technology*, *104*(9–12), 4617–4628.

Lin, Xuchao, & Cassandras, Christos G. (2015). An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces. *IEEE Transactions on Automatic Control*, *60*(6), 1659–1664.

Maini, Parikshit, Yu, Kevin, Sujit, P. B., & Tokekar, Pratap (2018). Persistent monitoring with refueling on a terrain using a team of aerial and ground robots. In *Proc. of IEEE intl. conf. on intelligent robots and systems* (pp. 8493–8498).

Maza, Iván, Caballero, Fernando, Capitán, Jesús, Martínez-De-Dios, J. R., & Ollero, Aníbal (2011). Experimental results in multi-UAV coordination for disaster management and civil security applications. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *61*(1–4), 563–585.

Menendez, Oswaldo, Auat Cheein, Fernando Alfredo, Perez, Marcelo, & Kouro, Samir (2017). Robotics in power systems: Enabling a more reliable and safe grid. *IEEE Industrial Electronics Magazine*, *11*(2), 22–34.

Miller, Kenneth S. (1981). On the inverse of the sum of matrices. *Mathematics Magazine*, *54*(2), 67.

Ng, Andrew Y., Jordan, Michael I., & Weiss, Yair (2001). On spectral clustering: Analysis and an algorithm. In *Proc. of 14th intl. conf. on neural information processing systems* (pp. 849–856).

Nilsson, Christian (2003). *Heuristics for the traveling salesman problem*: *Technical report*, Linkoping University, URL: https://www.researchgate.net/publication/228906083_Heuristics_for_the_Traveling_Salesman_Problem.

Pinto, Samuel C., Andersson, Sean B., Hendrickx, Julien M., & Cassandras, Christos G. (2020). Multi-agent infinite horizon persistent monitoring of targets with uncertain states in multi-dimensional environments. In *Proc. of 21st IFAC world congress*.

Rezazadeh, Navid, & Kia, Solmaz S. (2019). A sub-modular receding horizon approach to persistent monitoring for a group of mobile agents over an urban area. *IFAC-PapersOnLine*, *52*(20), 217–222.

Shen, Zhaolong, & Andersson, Sean B. (2011). Tracking nanometer-scale fluorescent particles in two dimensions with a confocal microscope. *IEEE Transactions on Control Systems Technology*, *19*(5), 1269–1278.

Shi, Jianbo, & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905.

Smith, Stephen L., Schwager, Mac, & Rus, Daniela (2011). Persistent monitoring of changing environments using a robot with limited range sensing. In *Proc. of IEEE intl. conf. on robotics and automation* (pp. 5448–5455).

Song, Cheng, Liu, Lu, Feng, Gang, & Xu, Shengyuan (2014). Optimal control for multi-agent persistent monitoring. *Automatica*, *50*(6), 1663–1668.

Sun, Chuangchuang, Welikala, Shirantha, & Cassandras, Christos G. (2020). Optimal composition of heterogeneous multi-agent teams for coverage problems with performance bound guarantees. *Automatica*, *117*, Article 108961. http://dx.doi.org/10.1016/j.automatica.2020.108961.

Trevathan, Jarrod, & Johnstone, Ron (2018). Smart environmental monitoring and assessment technologies (SEMAT)—A new paradigm for low-cost, remote aquatic environmental monitoring. *Sensors (Switzerland)*, *18*(7).

von Luxburg, Ulrike (2007). A tutorial on spectral clustering. arXiv e-prints, 0711.0189. URL: http://arxiv.org/abs/0711.0189, arXiv:0711.0189.

Welikala, Shirantha, & Cassandras, Christos G. (2019). Asymptotic analysis for greedy initialization of threshold-based distributed optimization of persistent monitoring on graphs. arXiv e-prints, 1911.02658. URL: http://arxiv.org/abs/1911.02658, arXiv:1911.02658.

Welikala, Shirantha, & Cassandras, Christos G. (2020). Asymptotic analysis for greedy initialization of threshold-based distributed optimization of persistent monitoring on graphs. *53*, In *Proc. of 21st IFAC World Congress* (2), (pp. 3433–3438). http://dx.doi.org/10.1016/j.ifacol.2020.12.1670.

Welikala, Shirantha, & Cassandras, Christos G. (2021). Event-driven receding horizon control for distributed persistent monitoring in network systems. *Automatica*, *127*, 109519. http://dx.doi.org/10.1016/j.automatica.2021.109519.

Yamashita, Atsushi, Arai, Tamio, Ota, Jun, & Asama, Hajime (2003). Motion planning of multiple mobile robots for cooperative manipulation and transportation. *IEEE Transactions on Robotics and Automation*, *19*(2), 223–237.

Yu, Jingjin, Karaman, Sertac, & Rus, Daniela (2015). Persistent monitoring of events with stochastic arrivals at multiple stations. *IEEE Transactions on Robotics*, *31*(3), 521–535.

Yu, Jingjin, Schwager, Mac, & Rus, Daniela (2016). Correlated orienteering problem and its application to persistent monitoring tasks. *IEEE Transactions on Robotics*, *32*(5), 1106–1118.

Zhou, N., Cassandras, C. G., Yu, X., & Andersson, S. B. (2019). Optimal threshold-based distributed control policies for persistent monitoring on graphs. In *Proc. of American control conf.* (pp. 2030–2035).

Zhou, Nan, Yu, Xi, Andersson, Sean B., & Cassandras, Christos G. (2018). Optimal event-driven multi-agent persistent monitoring of a finite set of data sources. *IEEE Transactions on Automatic Control*, *63*(12), 4204–4217.

Shirantha Welikala received the B.Sc. degree in electrical and electronic engineering from University of Peradeniya, Peradeniya, Sri Lanka, in 2015 and the M.Sc. and the Ph.D. degrees in systems engineering from Boston University, Brookline, MA, USA, in 2019 and 2021, respectively. From 2015 to 2017, he was with the Department of Electrical and Electronic Engineering, University of Peradeniya, Sri Lanka, where he worked first as a Temporary Instructor and subsequently as a Research Assistant. He is currently a Postdoctoral Research Fellow in the Department of Electrical Engineering, University of Notre Dame, South Bend, IN, USA. His main research interests include control and optimization of cooperative multi-agent systems with a particular emphasis on coverage and monitoring applications, robotics, signal processing, and smart grid applications. He is a recipient of several awards, including the 2015 Ceylon Electricity Board Gold Medal, the 2019 President's Award for Scientific Research in Sri Lanka and the 2021 Outstanding Ph.D. Dissertation Award in Systems Engineering.

Christos G. Cassandras is Distinguished Professor of Engineering at Boston University. He is Head of the Division of Systems Engineering, Professor of Electrical and Computer Engineering, and co-founder of Boston University's Center for Information and Systems Engineering (CISE). He received degrees from Yale University, Stanford University, and Harvard University. In 1982–84 he was with ITP Boston, Inc. where he worked on the design of automated manufacturing systems. In 1984–1996 he was a faculty member at the Department of Electrical and Computer Engineering, University of Massachusetts/Amherst. He specializes in the areas of discrete event and hybrid systems, cooperative control, stochastic optimization, and computer simulation, with applications to computer and sensor networks, manufacturing systems, and transportation systems. He has published about 450 refereed papers in these areas, and six books. He has guest-edited several technical journal issues and currently serves on several journal Editorial Boards, including Editor of Automatica. In addition to his academic activities, he has worked extensively with industrial organizations on various systems integration projects and the development of decision support software. He has most recently collaborated with The MathWorks, Inc. in the development of the discrete event and hybrid system simulator SimEvents.

Dr. Cassandras was Editor-in-Chief of the IEEE Transactions on Automatic Control from 1998 through 2009 and has also served as Editor for Technical Notes and Correspondence and Associate Editor. He was the 2012 President of the IEEE Control Systems Society (CSS). He has also served as Vice President for Publications and on the Board of Governors of the CSS, as well as on several IEEE committees, and has chaired several conferences. He has been a plenary/keynote speaker at numerous international conferences, including the 2017 IFAC World Congress, the American Control Conference in 2001 and the IEEE Conference on Decision and Control in 2002 and 2016, and has also been an IEEE Distinguished Lecturer.

He is the recipient of several awards, including the 2011 IEEE Control Systems Technology Award, the Distinguished Member Award of the IEEE Control Systems Society (2006), the 1999 Harold Chestnut Prize (IFAC Best Control Engineering Textbook) for Discrete Event Systems: Modeling and Performance Analysis, a 2011 prize and a 2014 prize for the IBM/IEEE Smarter Planet Challenge competition, the 2014 Engineering Distinguished Scholar Award at Boston University, several honorary professorships, a 1991 Lilly Fellowship and a 2012 Kern Fellowship. He is a member of Phi Beta Kappa and Tau Beta Pi. He is also a Fellow of the IEEE and a Fellow of the IFAC.