An MILP approach for persistent coverage tasks with multiple robots and performance guarantees

Charitidou, Maria; Keviczky, Tamás

DOI
10.1016/j.ejcon.2021.12.005

Publication date
2022

Document Version
Final published version

Published in
European Journal of Control

European Journal of Control 64 (2022) 100610

Contents lists available at ScienceDirect

European Journal of Control

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

An MILP approach for persistent coverage tasks with multiple robots and performance guarantees☆

Maria Charitidou [a,*], Tamás Keviczky [b]

[a] Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
[b] Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands

ARTICLE INFO

Article history:
Received 14 July 2020
Revised 24 December 2021
Accepted 27 December 2021
Available online 13 January 2022

Recommended by Prof. T Parisini

Keywords:
Multi-robot systems
MILP Planning
Persistent coverage
Receding horizon scheme

ABSTRACT

Multiple robots are increasingly being considered in a variety of tasks requiring continuous surveillance of a dynamic area, examples of which are environmental monitoring, and search and rescue missions. Motivated by these applications, in this paper we consider the multi-robot persistent coverage control problem over a grid environment. The goal is to ensure a desired lower bound on the coverage level of each cell in the grid, that is decreasing at a given rate for unoccupied cells. We consider a finite set of candidate poses for the agents and introduce a directed graph with nodes representing their admissible poses. We formulate a persistent coverage control problem as a MILP problem that aims to maximize the coverage level of the cells over a finite horizon. To solve the problem, we design a receding horizon scheme (RHS) and prove its recursive feasibility property by introducing a set of time-varying terminal constraints to the problem. These terminal constraints ensure that the agents are always able to terminate their plans in pre-determined closed trajectories. A two-step method is proposed for the construction of the closed trajectories, guaranteeing the satisfaction of the coverage level lower bound constraint, when the resulting closed trajectories are followed repeatedly. Due to the special structure of the problem, agents are able to visit every cell in the grid repeatedly within a worst-case visitation period. Finally, we provide a computational time analysis of the problem for different simulated scenarios and demonstrate the performance of the RHS problem by an illustrative example.

1. Introduction

Recent advances in the capabilities of robotic agents have led to an increase in the number of the tasks agents can perform as a team. Among others, particular tasks requiring repetitive execution have attracted interest, examples of which are area surveillance [34], cleaning [10] and forest fire monitoring [8]. In such tasks agents need to cooperatively plan their moves so that a given area is continuously covered. This problem is known in literature as the *Persistent Coverage Control* (PCC) problem [24].

PCC is closely related to the *Coverage Control* (CC) problem in which an area needs to be covered either once (*sweeping methods* [1,11,23,31,40]) or until a desired level of coverage is reached (*dynamic coverage methods*[12,14,29]). *Sweeping methods* are based on the decomposition of the area of interest in cells [11] assigned to the agents either online [31] or prior to the task. Contrary to previous approaches, in which coverage was achieved by non-moving agents optimally placed in each Voronoi cell of the area [6], here mobile agents are considered, covering the assigned cells by following a lawn-mowing pattern [23]. Other methods [1,40] approximate the area with a grid and construct *Spanning Trees* connecting the centers of the cells such that the maximum distance between any two agents is minimized.

In all these methods every point in the area is considered equally important for coverage. This assumption is relaxed in *Dynamic Coverage* methods [12,14,29], where each point has a value assigned to it expressing its coverage priority. Here, the agents are equipped with sensors with a known sensor model and the goal is to provide a desired level of coverage at fixed points in the area.

A common characteristic of the aforementioned CC tasks is their finite duration. This differentiates them from PCC tasks as the latter are executed repeatedly. In literature several solution approaches to the PCC problem are proposed that either decouple the sensor deployment problem from trajectory planning [27,36] or consider a motion planning framework under coverage specific objectives (*patrolling methods* [25,30,32], *coverage level methods*

☆ This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

* Corresponding author.
E-mail addresses: mariacha@kth.se (M. Charitidou), T.Keviczky@tudelft.nl (T. Keviczky).

https://doi.org/10.1016/j.ejcon.2021.12.005
0947-3580/© 2022 European Control Association. Published by Elsevier Ltd. All rights reserved.

[26,28,33,35,]). To overcome the complexity of PCC, authors in [27,36] propose a multi step method in which closed paths are designed for guaranteeing periodic coverage of points/sub-areas with known, predetermined periods while maintaining a desired level of coverage [27] or respecting frequency of visitation constraints [36]. In [4] stochastic Petri nets are considered for modelling the PCC problem under stochastic duration times of the coverage tasks while [3] proposes policies ensuring asymptotic satisfaction of a set of given specifications.

Patrolling methods consider a graph G with vertices defined as a finite set of points in the area to be covered and the goal is to design paths such that the maximum time elapsed since a point was last visited is minimized over the vertex set. In [30] polynomial time methods are proposed for the design of agents' paths in chain, tree and cyclic graphs. In [25] a greedy policy is introduced for non-holonomic agents while in [32] closed paths are designed considering different frequency of visitation constraints for each vertex of G. In all these methods, paths are designed once and possibly offline with the PCC task being considered successful, when agents follow these paths repeatedly.

While patrolling methods consider a static environment, in *coverage level* based methods the "level of coverage" of a point in the area to be covered is considered time-varying with known dynamics. Early works consider a finite set of points in the area and assume the existence of a single [19] or multiple closed paths [33] passing through every point in the set. Then, the goal is to design speed controllers so as the coverage level at these points is asymptotically driven to a desired value [19] or becomes asymptotically bounded [33]. *Infinitesimal Perturbation Analysis* was used in [20] for designing closed, elliptic trajectories minimizing the "coverage level loss" over a finite set of points in the area to be covered or more recently for allocating agents along linear segments minimizing coverage level loss while accounting for dwell times at the points of interest [41–43]. In [26,28] the coverage level of every point in the area is considered and the goal is to plan agents' actions so that a desired level of coverage is maintained over the area. To that end, authors propose feedback control laws steering agents towards less covered points in the area lying inside the agents' Voronoi cells. Nevertheless, authors do not consider frequency of visitation constraints in the area while cases of non-uniform coverage are allowed with the coverage level of some points being significantly lower than the desired one or close to zero.

In this paper we consider a grid environment and a team of robots responsible for maintaining a desired lower bound on the coverage level of each cell in the grid at all times. Similar to [26,28], the coverage level of each cell is decreasing over time for non-visited cells. Nevertheless, in our formulation we allow only one agent per time step to contribute to the coverage level of each cell by resetting it to a given constant. As a result, and due to the lower boundedness of the coverage levels, we can derive our first contribution as a lower bound on the frequency of visits at every cell. This allows us, in contrast to existing literature, to provide simultaneously both an upper bound on the worst-case revisiting time interval length and a lower bound on the quality of coverage of each cell. The second contribution of this paper is the formulation of a finite horizon MILP problem, the solution of which defines the trajectories of the agents satisfying the lower bound constraint over the horizon. To guarantee feasibility when the problem is solved in a receding horizon scheme (RHS), a set of time-varying terminal constraints is added to the problem. These constraints force agents' final poses to be along a set of predefined, closed trajectories and we propose a two step method for their design. To the best of our knowledge, this is the first online trajectory planning method with both worst-case frequency of visitation and coverage level guarantees.

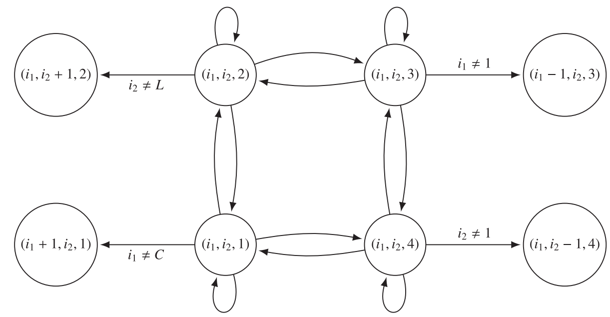


Fig. 1. Representation of the admissible moves for an arbitrary node $(x_{i_1}, y_{i_2}, \theta_{i_3})$. For simplicity we omit x, y, θ and use only the indices i_1, i_2, i_3.

The remainder of the paper is organized as follows: Section 2 introduces the basic elements of the problem. In Section 3 the MILP formulation of the problem is established. Section 4 presents the RHS-problem and the proposed two-step method for designing the closed trajectories for the terminal sets. Finally, in Section 5 numerical simulation results are shown while in Section 6 our conclusions are summarized and directions for future research are proposed.

2. Problem formulation

In this work a known, compact area $Q \subset \mathbb{R}^2$ is considered. The area is decomposed into a grid of $n_w = C \times L$ square cells with L, C denoting the number of rows and columns in the grid respectively. Without loss of generality a Cartesian coordinate system is assigned to the grid and an index $w \in I = \{1, \ldots, n_w\}$ to each cell in the grid. A team of n_r agents is employed for the task. The agents are equipped with identical sensors of finite and known sensing area.

Remark 1. Grid-based environments have been extensively considered in robotics for path planning, given a starting and goal position of an agent [2,16,37]. The grid is often abstracted by a graph with nodes representing the centers of the cells and edges connecting the neighboring cells, allowing the use of graph based algorithms such as Dijkstra and A* [15] for finding the shortest path with respect to a given cost function. Contrary to these approaches, where the size of the cells is often chosen arbitrarily small, in (persistent) coverage control problems a coarser grid can be generally considered, where the size of the cells is at most equal to the sensing area of the agents. This low resolution choice of grid offers several computational benefits while ensures that no point in the area is left uncovered [13].

Each agent is assigned to an index $r \in K = \{1, \ldots, n_r\}$ and its position and heading at time step k is denoted by p_r^k and θ_r^k respectively. In this work, the agents' allowable poses (positions and headings) are finite and agents' moves are constrained. At each time step k an agent is placed at the center of a cell $w \in I$, denoted by c_w with its heading θ_r^k taking values from the set $\left\{0, \pi, \frac{\pi}{2}, \frac{3\pi}{2}\right\}$. Based on their current position and heading, agents are able to perform one of the following actions:

- Stay at place (position and heading stays the same)
- Turn at place by 90°
- Move to an adjacent cell in the direction of their heading

Let the allowable poses of the agents be of the form $(x_{i_1}, y_{i_2}, \theta_{i_3})$, $i_1 \in \{1, \ldots, C\}$, $i_2 \in \{1, \ldots, L\}$, $i_3 \in \{1, \ldots, 4\}$ with $\theta_1 = 0$, $\theta_2 = \frac{\pi}{2}$, $\theta_3 = \pi$, $\theta_4 = \frac{3\pi}{2}$. Then, for arbitrary center coordinates (x_{i_1}, y_{i_2}) depending on the heading θ_{i_3}, $i_3 \in \{1, 2, 3, 4\}$ the admissible moves are defined as in Fig. 1.

To simplify notation we introduce a single-index label q to each allowable pose $(x_{i_1}, y_{i_2}, \theta_{i_3})$ with q defined as:

$$q = 4\,(i_1 - 1) + 4\,(i_2 - 1)\,C + i_3, \tag{1}$$

where $i_1 \in \{1, \ldots, C\}, i_2 \in \{1, \ldots, L\}, i_3 \in \{1, 2, 3, 4\}$. Based on the above definition the index of the cell w is expressed with respect to q by the following equation:

$$w = \left\lceil \frac{q}{4} \right\rceil.$$

Using the simplified labelling, we introduce a directed graph $G(V, E)$ with V the set of nodes q defined in (1) and E the set of admissible moves as presented in Fig. 1.

Next, we consider the coverage level at the center of cell $w \in I$ at time step k, denoted by $\mathbf{z}_k(w)$. Depending on the task, the coverage level may express the amount of dust removed in a cleaning task, the temperature in a heating task or the quality of information in an information gathering task. In [26,28] authors consider a non-negative coverage level value for each point that increases per time step with respect to the contribution of every agent at the corresponding point. Here, we allow at most one agent to contribute to the coverage level increase per time step. More specifically, we define the coverage level dynamics as follows:

$$\mathbf{z}_k(w) = d_w(1 - \sigma_w^k)\mathbf{z}_{k-1}(w) + \sigma_w^k \bar{Z}, \tag{2}$$

with $d_w \in (0, 1)$ a known, constant value called the *coverage decay factor*, $\bar{Z} > 0$ a known, constant value to which the coverage level resets when an agent visits the corresponding cell center and σ_w^k a binary variable representing occupancy of cell w by an agent, defined as:

$$\sigma_w^k = \begin{cases} 1, & \exists\, r \in K : \left\lceil \frac{q_r^k}{4} \right\rceil = w \\ 0, & \text{otherwise} \end{cases},$$

with q_r^k the pose of agent r at time instant k. Based on (2), the coverage level of cell w decreases with a rate proportional to the coverage decay factor d_w when w is not covered and resets to $\bar{Z}$ if an agent is placed at its center.

In many coverage tasks we may require agents to cover the area providing at least a sufficient level of coverage equal to a constant, pre-defined positive value $\underline{Z} < \bar{Z}$. Therefore, the following should hold for every k:

$$\mathbf{z}_k(w) \geq \bar{Z}. \tag{3}$$

Based on the above, we are now in a position to define the problem considered in this paper as follows:

Problem 1. Given a planning horizon of N time steps, a team of n_r agents, the graph G defining the allowable poses and moves of the agents and the coverage level dynamics defined by (2), design the agents' trajectories such that (3) is satisfied for all $k \in T_N = \{1, \ldots, N\}$.

If $N = \infty$ we will refer to Problem 1 as the *infinite horizon PCC* problem and consider $T_N \backslash \{N\} = T_N$.

3. Formulation of the MILP problem

In this section we formulate Problem 1 as a *Mixed Integer Linear Program* (MILP) that aims at maximizing the sum of the coverage levels of all cells while penalizing agents' intention to visit cells covered by peers at the previous time instant. A set of binary variables $x_{qq'}^k$ is introduced expressing whether a transition from pose $q \in V$ to $q' \in V$ is activated or not at time step k. Their exact definition is the following:

$$x_{qq'}^k = \begin{cases} 1, & \text{if at time } k\ \exists\, r \in K \text{ performing transition } (q, q') \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $(q, q') \in E,\ k \in T_N$.

The above variables capture the change on the agents' poses while at the same time remain agnostic about the agent performing the pose change. Moreover, their number depends on the size of the grid and the horizon length, i.e, it is independent of the size of the team.

In order to discourage distinct agents from covering the same cell at subsequent time instants, we introduce an extra set of binary variables μ_w^k, depending on $x_{qq'}^k$ such that the following constraints are satisfied:

$$\mu_w^k - \sum_{(q,q') \in V_w} x_{qq'}^k \leq 0, \tag{5a}$$

$$\mu_w^k - \sum_{(s,s') \in V_w \backslash V_w'} x_{ss'}^{k+1} \leq 0, \tag{5b}$$

$$\sum_{(q,q') \in V_w} x_{qq'}^k + \sum_{(s,s') \in V_w \backslash V_w'} x_{ss'}^{k+1} - \mu_w^k \leq 1, \tag{5c}$$

$\forall w \in I,\ k \in T_N \backslash \{N\}$ with $V_w = \big\{(q, q') \in E :\ q \in V,\ q' = 4(w-1) + 1, \ldots, 4w\big\}$, $V_w' = \big\{(q, q') \in E :\ q, q' = 4(w-1) + 1, \ldots, 4w\big\}$. Constraint (5a) ensures that $\mu_w^k \leq 1$ if there exists an agent r_1 covering w at time step k. Similarly, (5b) implies that $\mu_w^k \leq 1$ if at time step $k+1$ another agent r_2 enters w. If the agents r_1, r_2 visit w at time step k and $k+1$ respectively, by (5c) we have $\mu_w^k \geq 1$. Hence, $\mu_w^k = 1$. In all other cases $\mu_w^k = 0$. In many coverage applications it is often desirable to minimize unnecessary visits at cells in order to save resources and minimize costs. Therefore, as we discuss later in (11), the cost of subsequent visits at each cell w is linearly introduced with respect to μ_w^k and penalized over the horizon N.

In addition to the aforementioned binary variables, we define the continuous variables $\mathbf{z}_k(w) \in [0, \bar{Z}], w \in I, k \in T_N$ expressing the coverage level of cell w at time step k.

A direct consequence of the variable definitions stated above is the quadratic form of the coverage dynamics in (2), due to multiplication of decision variables. Addressing this problem, we reformulate the coverage level dynamics using a set of linear inequalities as follows:

$$-\mathbf{z}_k(w) + \bar{Z} \sum_{(q,q') \in V_w} x_{qq'}^k \leq 0, \tag{6a}$$

$$\mathbf{z}_k(w) - d_w \mathbf{z}_{k-1}(w) - \bar{Z} \sum_{(q,q') \in V_w} x_{qq'}^k \leq 0, \tag{6b}$$

$$\mathbf{z}_k(w) - d_w \mathbf{z}_{k-1}(w) - (1 - d_{\max})\bar{Z} \sum_{(q,q') \in V_w} x_{qq'}^k \geq 0, \tag{6c}$$

$$\mathbf{z}_k(w) \leq \bar{Z}, \tag{6d}$$

with $w \in I, k \in T_N$ and $d_{\max} = \max_{w \in I} d_w$. Due to (4), $\sigma_w^k = \sum_{(q,q') \in V_w} x_{qq'}^k$, for every $k \in T_N$ and $w \in I$. If $x_{qq'}^k = 0$ for all $(q, q') \in V_w$ (e.g., no agent visits w at time step k), then due to (6b) and (6c) we have:

$$\mathbf{z}_k(w) = d_w \mathbf{z}_{k-1}(w).$$

On the other hand, if $\sigma_w^k = 1$, then, due to (6a), (6d) it holds that $\mathbf{z}_k(w) = \bar{Z}$. Therefore, satisfaction of (6a)-(6d) ensures satisfaction of the equality constraint (2).

An important consideration for the current problem formulation is guaranteeing that the variables $x^k_{qq'}, k \in T_N$ uniquely define the trajectory of each agent over the horizon. To achieve this, we introduce the following constraint:

$$\sum_{(q,q')\in V_w} x^k_{qq'} \leq 1, \tag{7}$$

$w \in I, k \in T_N$. This guarantees that each cell is visited by at most one agent per time step. Therefore, given the pose transition history, starting at $k = N$ and back-propagating in time returns the initial poses of the agents. Given that agents are initialized at different cells, we may conclude the unique correspondence between trajectories and agents.

The aforementioned result is partially based on the fact that agents' pose transitions are admissible and exist over the horizon. To ensure the above, we consider the following constraints:

$$\sum_{q\in V}\sum_{q'\in V} x^k_{qq'} = n_r, \quad k \in T_N, \tag{8a}$$

$$x^k_{qq'} - \sum_{(q',q'')\in E} x^{k+1}_{q'q''} \leq 0, \quad k \in T_N \backslash \{N\}, (q,q') \in E. \tag{8b}$$

Constraint (8a) ensures that the number of pose transitions performed over the graph G at each time step $k \in T_N$ is equal to the number of agents in the team. Additionally, by (8b), if $x^k_{qq'} = 1$, then there should exist at least one variable $x^{k+1}_{q'q''}$ with $(q', q'') \in E$ such that $x^{k+1}_{q'q''} = 1$. This implies that an agent must perform a transition from pose q' to pose q'' at time step $k + 1$ only if $(q', q'') \in E$, i.e., only if the transition (q', q'') is admissible. Finally, constraints are introduced to define the initial pose transitions of the agents and initial coverage level of the cells. In addition, we consider (3) over the horizon and ensure that the problem variables are taking values among the admissible. These are summarized by the constraints below:

$$\sum_{(q_r,q)\in E} x^1_{q_r q} = 1, \qquad r \in K, \tag{9a}$$

$$\mathbf{z}_0(w) = \bar{Z}, \qquad w \in I, \tag{9b}$$

$$\mathbf{z}_k(w) \geq \bar{Z}, \qquad w \in I, k \in T_N, \tag{9c}$$

$$x^k_{qq'} \in \{0, 1\}, \quad (q, q') \in E, k \in T_N, \tag{9d}$$

$$\mu^k_w \in \{0, 1\}, \quad w \in I, k \in T_N \backslash \{N\}, \tag{9e}$$

where $q_r \in V$ is the initial pose of agent $r, r \in K$.

Let $\mathbf{x} = \begin{bmatrix}\mathbf{b}^T & \mathbf{z}^T & \boldsymbol{\mu}^T\end{bmatrix}^T$ be the variable vector with $\mathbf{b}$ the stacked vector of $x^k_{qq'}, (q, q') \in E, k \in T_N$, $\mathbf{z}$ the stacked vector of $\mathbf{z}_k(w), w \in I, k \in T_N$ and $\boldsymbol{\mu}$ the stacked vector of $\mu^k_w, w \in I, k \in T_N \backslash \{N\}$. In addition, let $\bar{n}_b, \bar{n}_z, \bar{n}_\mu$ be the length of the vector $\mathbf{b}, \mathbf{z}, \boldsymbol{\mu}$ respectively with $\bar{n}_b = (n_w - 2(C + L))N$, $\bar{n}_z = n_w N$ and $\bar{n}_\mu = n_w(N - 1)$. Then, we are in position to define the problem considered in this paper as follows:

$$\max_{\mathbf{x}} J(\mathbf{x}) \tag{10}$$

subject to the constraints:

$$\begin{aligned} &(5a) - (5c)\\ &(6a) - (6d)\\ &\qquad (7)\\ &(8a) - (8b)\\ &(9a) - (9e) \end{aligned}$$

with $J(\mathbf{x})$ the objective function of the problem defined as:

$$J(\mathbf{x}) = \begin{bmatrix}\mathbf{0}_{\bar{n}_b} & \mathbf{1}_{\bar{n}_z} & -\beta\mathbf{1}_{\bar{n}_\mu}\end{bmatrix}\mathbf{x}, \tag{11}$$

where $\mathbf{1}_\rho = \begin{bmatrix}1 & 1 & \ldots & 1\end{bmatrix}$ is the vector of 1s of length ρ and β a positive constant expressing the importance/weight of forcing agents to avoid cells covered at the previous time step by peers.

In the above problem revisiting frequency constraints are not explicitly defined. However, based on the coverage level dynamics and the required lower bound of the coverage level of each cell we can state the following:

Proposition 1. *If problem (10) is feasible, then the resulting trajectories guarantee a lower bound f_w on the frequency at which each cell w, $w \in I$ is visited, defined as:*

$$f_w = \left\lfloor \frac{\ln \bar{Z} - \ln \bar{Z}}{\ln d_w} \right\rfloor^{-1}.$$

Proof. The coverage level of w evolves over time based on (2) as follows:

$$\mathbf{z}_k(w) = d_w(1 - \sigma^k_w)\mathbf{z}_{k-1}(w) + \sigma^k_w \bar{Z}.$$

Due to (2), $\mathbf{z}_k(w)$ is monotonically decreasing between two subsequent visits at the cell. Assuming that an agent covers cell w at time step k_1,we have that $\mathbf{z}_{k_1}(w) = \bar{Z}$. We need (3) to hold until the next time an agent visits w, thus:

$$d^k_w \bar{Z} \geq \bar{Z}, \qquad k \geq k_1.$$

Solving the inequality above leads to: $k \leq \frac{\ln \bar{Z} - \ln \bar{Z}}{\ln d_w}$ and since $k \in \mathbb{Z}$ we have the result. □

4. Modified MILP with guaranteed feasibility

The MILP problem presented above guarantees a desired lower bound on the coverage level of the cells for N time steps. However, PCC is by its nature an infinite horizon problem, in which agents' actions need to be continuously planned so as the total coverage level of the area is maximized. As solving the infinite horizon PCC problem is computationally intractable, motivated by Model Predictive Control schemes [22], our approach is to implement the finite horizon solution of the problem described in Section 3, in a receding horizon fashion: each agent implements their first move from the solution of (10), then resolves the problem over a shifted time horizon in the next step, starting from their new pose. However, for $N < \infty$ problem (10) might not be always feasible due to different initial conditions, especially when the problem is solved recursively. In order to address this problem, in this Section we propose a modified version of (10) in which a set of time-varying terminal constraints is added to the problem as in [17]. These constraints force agents to move at the end of the prediction horizon along predefined, closed trajectories that are designed to guarantee the lower coverage level bound when repeatedly followed. In that way, it is possible to prove the recursive feasibility property of the new problem when it is solved under the receding horizon scheme.

Remark 2. Model Predictive Control has been extensively considered for setpoint stabilization [9] and reference tracking [19]. The problem involves the solution of a finite horizon optimization problem online and under certain assumptions can ensure stability of the system under consideration. Although initially proposed for deterministic systems, it can be efficiently applied to uncertain systems with bounded or unbounded noise ensuring asymptotic convergence to a neighborhood of the goal position or reference trajectory by means of an offline designed controller [18] or constraint tightening techniques [21].

Let $\mathbf{u} \in V^{n_r}$ be the stacked vector of the agents' current poses and $\mathbf{z}^{\mathbf{u}} \in [\bar{Z}, \bar{Z}]^{n_w}$ the vector of the coverage levels of the cells defined by (2) based on $\mathbf{u}$. For $M \in \mathbb{N},\ M > 0$ consider a sequence of sets $\{S_0, \dots, S_{M-1}\}$ with M the length of the closed trajectories and $S_\nu,\ \nu = 0, \dots, M-1$ defined as:

$$S_\nu = \left\{ \begin{bmatrix} \mathbf{z}^{\mathbf{u}} \\ \mathbf{u} \end{bmatrix} \in [\bar{Z}, \bar{Z}]^{n_w} \times V^{n_r} : \mathbf{z}^{\mathbf{u}} \succeq \mathbf{z}^{\mathbf{u}}_\nu, \mathbf{u} = \mathbf{u}_\nu \right\}, \tag{12}$$

where $\mathbf{z}^{\mathbf{u}}_\nu$ is the vector including the coverage levels of the cells at time step ν when the agents' poses are defined as elements of the vector $\mathbf{u}_\nu$. The sets S_ν are defined such that $\bigcup_{\nu=0}^{M-1} S_\nu$ introduces the set of the agents' closed trajectories. Therefore, it holds that:

$$\begin{cases} (\mathbf{u}^r_\nu, \mathbf{u}^r_{\nu+1}) \in E, & \nu \in \{0, \dots, M-2\} \\ (\mathbf{u}^r_{M-1}, \mathbf{u}^r_0) \in E \end{cases}, \tag{13}$$

for every $r \in K$ with $\mathbf{u}^r_\nu \in V$ defining the pose of agent r at time step ν. Due to (13), $\mathbf{z}^{\mathbf{u}}_{\nu+1}(w)$ is defined by (2) based on $\mathbf{z}^{\mathbf{u}}_\nu(w), \mathbf{u}_{\nu+1}$ for every $\nu \in \{0, \dots, M-2\}, w \in I$ and $\mathbf{z}^{\mathbf{u}}_0(w)$ based on $\mathbf{z}^{\mathbf{u}}_{M-1}(w), \mathbf{u}_0$. A systematic procedure on how to design these trajectories is presented later in Section 4.1.

Let $t_i \geq 0, i = 0, 1, \dots$ denote the absolute time instants at which the optimization problem is solved. The team is assumed to be capable of communicating with a central base or a specific agent responsible for solving the MILP problem. At each time instant t_i agents communicate with the base/agent receiving information about their planned trajectories. Each agent performs the first pose transition along its recently planned trajectory and the procedure is repeated over a shifted planning horizon.

Let $X_i^f \in \{S_0, \dots, S_{M-1}\}$ be the terminal constraint set at time t_i. The coverage level constraints defining X_i^f in (12) are inherently linear and thus can be directly incorporated to the modified MILP as constraints of the form:

$$\mathbf{z}_N(w) \geq \mathbf{z}^{\mathbf{u}}_\nu(w), \qquad w \in I. \tag{14}$$

On the other hand, $\mathbf{u} = \mathbf{u}_\nu$ is not linear in $\mathbf{x}$. A naive but straightforward way to introduce the final pose constraints linearly could be the following:

$$\sum_{(q, q_\nu) \in E} x^N_{q\, q_\nu} \geq 1, \qquad \forall q_\nu, \tag{15}$$

where $q_\nu \in V$ is an element of $\mathbf{u}_\nu,\ \nu \in \{0, \dots, M-1\}$. Although the above constraint forces agents' poses to be along one of the predefined closed trajectories, it does not guarantee that agents will have the same initial and final pose. For example, this could happen when two agents r_1, r_2 with initial poses $q_{r_1}, q_{r_2} \in V$ "flip" poses at the end of the horizon. To avoid this and drawing inspiration from [7], we introduce an extra set of integer variables denoted by $\kappa^k_w,\ w \in I, k \in T_N$. These variables take values in $K \cup \{0\}$ and are responsible for keeping track of the agents' moves over the horizon. If the cell w is covered by agent r at time step k, $\kappa^k_w = r$. Otherwise, we set $\kappa^k_w = 0$. This is equivalent to the satisfaction of the following constraints:

$$\kappa^k_w - n_r \sum_{(q, q') \in V_w} x^k_{qq'} \leq 0, \tag{16a}$$

$$\kappa^{k-1}_w - \kappa^k_{w'} + n_r x^k_{qq'} \leq n_r, \qquad w \neq w', \tag{16b}$$

$$\kappa^{k-1}_w - \kappa^k_w + n_r \sum_{(q, q') \in V'_w} x^k_{qq'} \leq n_r, \tag{16c}$$

$$\sum_{w=1}^{n_w} \kappa^k_w = \frac{n_r(n_r + 1)}{2}, \tag{16d}$$

$$\kappa^k_w \in K \cup \{0\}. \tag{16e}$$

Constraints (16a), (16e) guarantee $\kappa^k_w = 0$ for all cells $w \in I$ not covered at k while (16b)-(16c) introduce n_r inequalities of the form $\kappa^k_{w_c} \geq r_c,\ c = 1, \dots, n_r$. In order for these inequalities to be true in conjunction with (16d), (16e) $\kappa^k_{w_c} = r_c,\ c = 1, \dots, n_r$ should hold. Thus, the new variables are well-defined.

Considering the newly introduced variables in the modified MILP, we may define their initial and final conditions within the problem as follows:

$$\kappa^0_w = \begin{cases} r, & w = \left\lceil \frac{q_{r,t_i}}{4} \right\rceil \\ 0, & \text{otherwise} \end{cases}, \tag{17a}$$

$$\kappa^N_w = \begin{cases} r, & w = \left\lceil \frac{\mathbf{u}^r_\nu}{4} \right\rceil \\ 0, & \text{otherwise} \end{cases}, \tag{17b}$$

where $q_{r,t_i} \in V$ is the pose of agent r at time t_i and $\mathbf{u}^r_\nu$ is the r-th element of the vector $\mathbf{u}_\nu$.

Let $\hat{\mathbf{x}} = \begin{bmatrix} \mathbf{x}^T & \boldsymbol{\kappa}^T \end{bmatrix}^T$ be the new decision variable vector where $\boldsymbol{\kappa}$ is the stacked vector of the κ^k_w variables of length $\bar{n}_\kappa = n_w N$. At each time instant t_i the base/agent in charge updates the poses of the agents and the coverage level of the cells such that $q_r = q_{r,t_i},\ r \in K$ and $\mathbf{z}_0(w) = \mathbf{z}_{t_i}(w), w \in I$ and proceeds with the solution of the *modified-MILP* problem defined as:

$$\max_{\hat{\mathbf{x}}} J'(\hat{\mathbf{x}}) \tag{18}$$

subject to:

$$\begin{aligned} &(5a) - (5c) \\ &(6a) - (6d) \\ &(7) \\ &(8a) - (8b) \\ &(9a) - (9e) \\ &(14) - (15) \\ &(16a) - (16e) \\ &(17a) - (17b) \end{aligned}$$

with $\mathbf{z}_{t_i}(w) \in [\bar{Z}, \bar{Z}]$ the coverage level fo cell w at time t_i and $J'(\hat{\mathbf{x}}) = \begin{bmatrix} \mathbf{0}_{\bar{n}_b} & \mathbf{1}_{\bar{n}_z} & -\beta \mathbf{1}_{\bar{n}_\mu} & \mathbf{0}_{\bar{n}_\kappa} \end{bmatrix} \hat{\mathbf{x}}$ the objective function of the modified problem for which $J'(\hat{\mathbf{x}}) = J(\mathbf{x})$ holds.

At time t_0 the terminal constraint set X_0^f can be chosen as any set of the sequence such that (18) is feasible. Suppose $X_0^f = S_\nu$. It follows that $X_1^f = S_{\nu+1},\ X_2^f = S_{\nu+2},\ \dots,\ X^f_{M-1-\nu} = S_{M-1},\ X^f_{M-\nu} = S_0, \dots,\ X^f_{M-1} = S_\nu$. Then, for any $i \in \mathbb{N}$ we can obtain the following rule [17]:

$$X_0^f = S_\nu \Rightarrow X_i^f := S_{(\nu+i) \bmod M}. \tag{19}$$

Based on the above we can state the following theorem:

Theorem 1. *Suppose problem (18) is feasible at time t_i with initial coverage level values $\mathbf{z}_{t_i}(w), w \in I$, initial agents' poses $q_{r,t_i}, r \in K$ and terminal set X_i^f as defined in (19). Suppose $\mathbf{u}^*(t_i) = \begin{bmatrix} \mathbf{u}^*_{t_i+1} & \dots & \mathbf{u}^*_{t_i+N} \end{bmatrix} \in V^{n_r \times N}$ is a feasible sequence of the agents' poses found as a solution of (18) at time t_i. Then, the problem will be feasible at t_{i+1} with the initial poses of the agents defined by $\mathbf{u}^*_{t_i+1}$ and the initial coverage level of every cell w computed by (2) based on $\mathbf{z}_{t_i}(w)$ and $\mathbf{u}^*_{t_i+1}$.*

Proof. For any $t_i \in \mathbb{N}$ there exists an index $p \in \{0, \dots, M-1\}$ such that: $\begin{bmatrix} \mathbf{z}^*_{t_i+N} \\ \mathbf{u}^*_{t_i+N} \end{bmatrix} \in X_i^f = S_{(\nu+i) \bmod M} = S_p$ where $\mathbf{z}^*_{t_i+N}, \mathbf{u}^*_{t_i+N}$ are the vectors including the feasible poses of the agents and the feasible coverage levels of the cells at $k = t_i + N$ respectively. Due to (19) it holds that:

$$X_{i+1}^f = \begin{cases} S_{p+1}, & p < M-1 \\ S_0, & p = M-1 \end{cases}.$$

In order for the problem to be feasible at t_{i+1} a pose sequence $\mathbf{u}(t_{i+1}) \in V^{n_r \times N}$ should be defined such that: $\begin{bmatrix} \mathbf{z}_{t_{i+1}+N} \\ \mathbf{u}_{t_{i+1}+N} \end{bmatrix} \in X_{i+1}^f$. Consider the input sequence $\mathbf{u}(t_{i+1}) = \begin{bmatrix} \mathbf{u}_{t_i+2}^* & \cdots & \mathbf{u}_{t_i+N}^* & \mathbf{u} \end{bmatrix}$ with:

$$\mathbf{u} = \begin{cases} \mathbf{u}_{p+1}, & p < M-1 \\ \mathbf{u}_0, & p = M-1 \end{cases}.$$

Due to (13), $(\mathbf{u}_{t_i+N}^{*\,r}, \mathbf{u}^r) \in E$ for any r. In addition, $\mathbf{z}_{t_{i+1}+N}(w)$ is computed by (2) based on $\mathbf{z}_{t_i+N}^*(w), \mathbf{u}$ for all $w \in I$. For the cells covered at $t_{i+1}+N$ both the coverage levels at $t_{i+1}+N$ and the coverage levels of the terminal set corresponding to these cells are equal to $\bar{Z}$. For the rest of the cells due to the linear coverage level dynamics of (2) and the construction of the sets, it holds that $\mathbf{z}_{t_{i+1}+N}(w) = d_w\, \mathbf{z}_{t_i+N}^*(w) \geq d_w\, \mathbf{z}_p^{\mathbf{u}}(w) = \mathbf{z}_{p'}^{\mathbf{u}}(w)$ with $p' = p+1$ if $p < M-1$ or 0 otherwise. Hence, $\begin{bmatrix} \mathbf{z}_{t_{i+1}+N} \\ \mathbf{u}_{t_{i+1}+N} \end{bmatrix} \in X_{i+1}^f$ holds. This completes the proof. □

4.1. Designing the terminal trajectories

An important question arising at this point is how to design the closed trajectories guaranteeing the recursive feasibility of (18). Towards this goal, we propose a two-step method for designing a set of closed trajectories for the agents. These trajectories are jointly constructed to ensure satisfaction of (3) at all times when agents follow their corresponding trajectories repeatedly. In the first step of the method the closed trajectories are found as the solution to an optimization problem of the form (18) in which we make the following modifications: 1) discard the terminal coverage level constraints (14) for every $w \in I$, 2) introduce constraints ensuring that the initial and terminal poses of the agents are the same and the corresponding variables κ_w^k at time steps $k = 0$ and $k = M$ satisfy $\kappa_w^M = \kappa_w^0$ and 3) add the following constraint to the problem:

$$\sum_{k=1}^{M} \sum_{(q,q') \in V_w} x_{qq'}^k \geq 1, \quad w \in I$$

guaranteeing that each cell of the grid will be covered at least once over the planning horizon of length M. The resulting trajectories guarantee the satisfaction of (3) when they are followed by the corresponding agents once. However, (3) might be violated when the same trajectories are repeatedly followed. To resolve this issue, we consider a second step and design a *Linear Program* (LP) is designed aiming at finding the minimum $\bar{Z}$ value for which (3) is always satisfied when the trajectories of step 1 are followed infinitely many times. This problem is of the following form:

$$\min \bar{Z} \tag{20}$$

subject to:

$$\mathbf{z}_k(w) = d_w(1-\sigma_w^k)\mathbf{z}_{k-1}(w) + \sigma_w^k \bar{Z}, \ k \in T_M, w \in I \tag{20a}$$

$$\sigma_w^k = \begin{cases} 1, & \exists\, r \in K : \left\lceil \frac{\mathbf{u}_k'^{\,r}}{4} \right\rceil = w \\ 0, & \text{otherwise} \end{cases}, k \in T_M, w \in I \tag{20b}$$

$$\mathbf{u}_k' = \mathbf{u}_{k,s}, \quad k \in T_M \tag{20c}$$

$$\mathbf{z}_0(w) = \mathbf{z}_{M,s}(w), \quad w \in I \tag{20d}$$

$$\mathbf{u}_0' = \mathbf{u}_{M,s}, \tag{20e}$$

$$\mathbf{z}_k(w) \in [\bar{Z}, \bar{Z}], \quad k \in T_M, w \in I, \tag{20f}$$

where $T_M = \{1, \ldots, M\}$ and $\mathbf{u}_{k,s}$, $\mathbf{z}_{k,s}$ the vector of the agents' poses and the vector of the coverage level of the cells at time k as found in step 1, respectively. The optimization problems at step 1 and 2 are solved over the same, fixed horizon M. If for a given M the MILP problem at step 1 is not feasible or a maximum computational time limit is reached, a larger M can be chosen [5]. If M reaches a maximum expected value, then it is possible that $\bar{Z}$ can not be ensured with the given number of agents n_r. Hence, the designer should consider increasing the number of agents in the team or decreasing the worse case visitation frequency of each cell by increasing the value $\bar{Z}$ at step 1.

5. Simulations

In this section we examine the computational performance of (10) under different scenarios and validate the effectiveness of the method presented in Section 4 with an illustrative example. All simulations were run on an Intel Xeon W-2145 3.70 GHz CPU, 31GB RAM computer using MATLAB 2018b while the MILP problems are solved using the commercial solver GUROBI 8.0.1. In the following simulation experiments a computational time-limit is set in GUROBI equal to 5 hours.

5.1. Computational time analysis

For the computational complexity analysis we consider a closed environment of size 24×24 m^2 decomposed into a 6×6 grid with 4 agents. The agents' trajectories are planned over a horizon of $N = 10$ steps, unless otherwise stated. Different maps of coverage decay factors are considered with the average decay factor value being lower bounded by 0.85. We set $\bar{Z} = 300$, $\bar{Z} = 20$, $\beta = 0.8$. Keeping the other elements of the problem unaltered, we consider changing one of the following: 1) the team size, 2) the planning horizon length and 3) the grid resolution when 20 different decay factor maps are given.

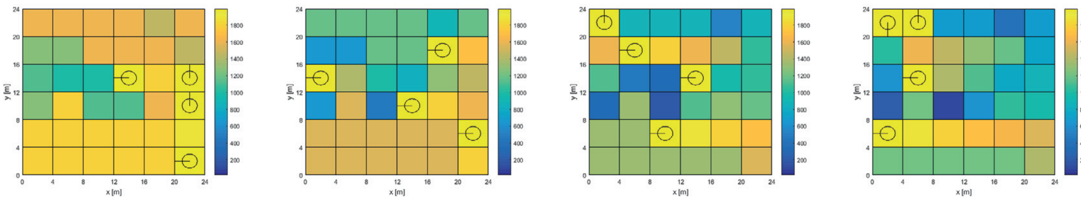
In the first experiment we consider teams of size $n_r = 3, 4, 5$ with the agents initially placed at cells $w_i = 6i, i \in K$ with headings equal to π. An example for $n_r = 4$ is shown in Fig. 4. In the second experiment, 4 agents are employed for the task and their trajectories are planned over an optimization horizon of $N = 7, 10, 12$ time steps. In all these experiments we assume that the environment is already decomposed into a 6×6 grid. In the last experiment, we introduce different grid resolutions and plan the 4 agents' trajectories over a planning horizon $N = 10$. Here, we consider a grid resolution $\alpha \times \alpha$ with $\alpha = 4, 6, 8$. Let $\mathcal{L}_\alpha(w_\alpha^i)$ denote the set of points $(x, y) \in \mathbb{R}^2$ belonging to the i-th cell of the grid of resolution $\alpha \times \alpha$ with $i \in I_\alpha, I_\alpha = \{1, \ldots, \alpha^2\}$. In addition, let us introduce the set $\mathcal{J}_\alpha(w_\alpha^i)$ as:

$$\mathcal{J}_\alpha(w_\alpha^i) = \left\{ j \in I_6 : \mathcal{L}_\alpha(w_\alpha^i) \cap \mathcal{L}_6(w_6^j) \neq \emptyset \right\}$$

This set includes the indices of the cells in the 6×6 grid "sharing" points $(x, y) \in \mathbb{R}^2$ with w_α^i. Based on that, we may define the decay factor $d_{w_\alpha^i}$ of the cell $w_\alpha^i, i \in I_\alpha$ in the $\alpha \times \alpha$ resolution as follows:

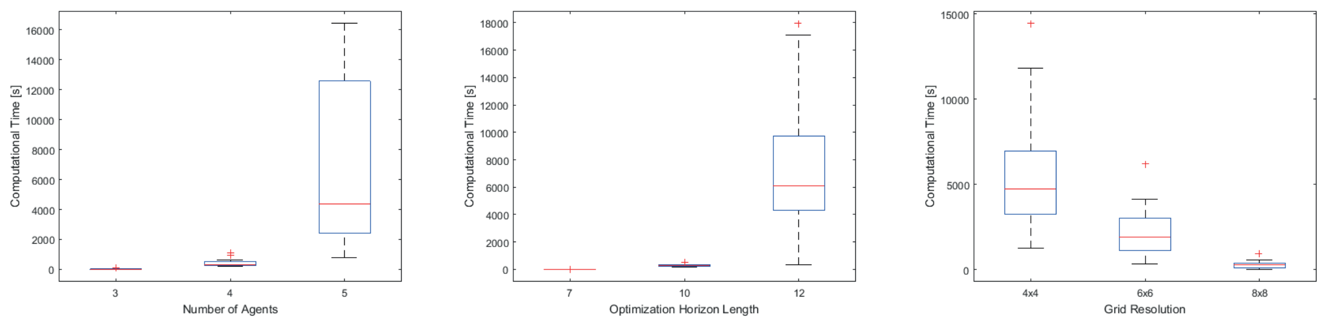
$$d_{w_\alpha^i} = \min \left\{ d_{w_6^j} : j \in \mathcal{J}_\alpha(w_\alpha^i) \right\} \tag{21}$$

for $i \in I_\alpha$, $\alpha = 4, 8$. The agents are initially placed at cells with indices $i\alpha, i = 1, \ldots, 4$ with headings equal to π. As found in practice $\bar{Z} = 300$ renders many problems infeasible. Therefore, only for the last experiment $\bar{Z}$ is chosen equal to 800 as this is the smallest value guaranteeing convergence for all 20 cases within the time limit independent of the choice of resolution.



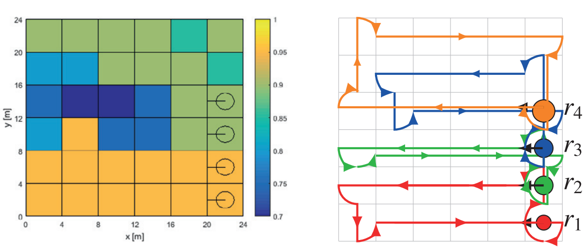
(a) $i = 2$ (b) $i = 5$ (c) $i = 8$ (d) $i = 10$

Fig. 2. Coverage Level Map and Agents' Poses at different time instants t_i over the simulation horizon.



(a) (b) (c)

Fig. 3. Computational Time to Convergence for a varying (a) number of agents, (b) planning horizon length and (c) grid resolution when coverage level maps with average decay factor $\bar{d}_w \geq 0.85$ are considered. In (3a) a 6×6 grid is considered and paths are designed over a horizon $N = 10$. In (3b) we consider 4 agents working on a 6×6 grid while in (3c) a team of 4 agents is employed and paths are designed over an optimization horizon $N = 10$.



(a) (b)

Fig. 4. The coverage decay map and the constructed terminal, closed trajectories for Scenario 1 a) the map of the coverage decay factors d_w b) the resulting terminal trajectories.

Number of agents In Fig. 3 the computational time required by the solver to terminate is presented in seconds. When different team sizes are considered (Fig. 3a) 80% of the cases achieve convergence before the time limit with the average time increasing from 8 to 4100 sec when n_r is 3 and 5 respectively. This is a direct consequence of the increased number of the cell combinations when the number of agents and consequently the possible transitions increase. As the number of agents increases, the number of non-zero variables $x^k_{qq'}$ increases proportionally allowing more cells to be covered per time step. Therefore, the MILP solver, often based on a branch and cut method, may require more time to expand the tree of possible solutions, evaluate their feasibility and cost with respect to the objective function, and possibly cut those that are found non-feasible or costly with respect to the given solution.

Planning horizon Similarly, an increasing computational time is observed in Fig. 3b when the horizon length grows with the average time for convergence being 6100 sec for $N = 12$. This increase is partially expected due to the proportional relationship of N with the number of binary variables in the problem. We also note that only 50% of the cases converge within the predefined time limit highlighting the difficulty of the solver to cut infeasible nodes and move towards the parts of the decision tree maximizing the objective function.

Number of grid cells In the final experiment, a different relation between the computational time and number of grid cells is shown. When $n_w = 64$ the average time for convergence is 290 sec with the latter reaching 4800 sec for $n_w = 16$. This result may be explained with respect to the number of cells with low decay factors and the (minimum) distance between the agents and the centers of these cells over the graph G. On the one hand, due to (21) the number of cells w with low d_w factors increases both in the 4×4 and the 6×6 resolution of the grid. It is indicative that only 7.64% of the cells in average has a worst case revisiting interval length ≤ 10 in the 6×6 grid with this amount rising to 29.36% and 15.78% in the 4×4 and 8×8 grid respectively (the number of the cells with $f_w^{-1} \leq 10$ is averaged over the total number of cells of the 20 different coverage decay maps of this study). On the other hand, agents may cover every cell in the 4×4 grid once after 4 time steps. This gives them the freedom to visit several different cells over the horizon with the solver requiring more time to examine which combination of plans is the most coverage-effective. On the contrary, agents should choose wisely which cells to visit in the 8×8 grid so as (3) is always satisfied. This may often mean that agents will reach cells with low decay factors exactly after f_w^{-1} time steps due to the large number of edges in G to be traversed. Hence, the number of feasible plans in the 8×8 grid is significantly reduced compared to the 4×4 case resulting in the surprising computational time result shown in Fig. 3c.

In general, we may conclude that the computational time increases with the number of agents and the length of the horizon while the results with respect to the grid resolution are amenable to the choice of the decay factors. Nevertheless, as the resolution of the grid becomes finer, we expect an increase in the computational time when more coverage decay factor maps are added to the study. For example, the complexity of the problem may increase in the following cases: 1) when for every cell w, under the reset value $\bar{Z}$ the worst case visitation period of w satisfies $f_w^{-1} \gg 10$, or

2) when the reset value $\bar{Z}$ becomes higher, increasing f_w^{-1}, for every $w \in I$.

Future work will consider ways to decrease the computational complexity of the problem. A promising way towards reducing the "curse of dimensionality" is the design of a distributed framework in which agents will design their plans based on local information exchange among a small number of peers. When large areas are considered, the problem could also benefit from an initial decomposition of the area and an offline assignment of regions to small teams of agents. Finally, with the rapid development of 5G technologies and cloud services, the proposed problem could be also solved off-board by remote, powerful servers [39] while accounting for problems related to the quality of communication and the latency.

5.2. Performance of the RHS scheme

To validate the feasibility of the modified MILP of Section 4, we simulate a persistent coverage task in a grid environment of 6×6 square cells with 4 agents. We refer to this example as Scenario 1. The lower coverage level $\underline{Z}$ is set equal to 20 and $\beta = 0.8$. The coverage decay factors of the cells and the initial poses of the agents are shown in Fig. 4a. Initially, we design the closed trajectories for guaranteeing feasibility. The planning horizon of the closed trajectories is chosen arbitrarily to be $M = 18$. The trajectories found in step 1 do not satisfy (3) when repeatedly followed. Therefore, step 2 is initiated and $\bar{Z}$ is found as the solution of (20) and equal to 1995.5. The constructed trajectories are shown in Fig. 4b.

Given the closed trajectories and $\bar{Z} = 1995.5$, we run (18) for a simulation horizon of 100 steps. The optimization horizon N is set equal to 18. Initially, the terminal constraint set X_0^f is chosen to be S_0. In Fig. 2 the coverage level map of the grid and the corresponding poses of the agents are shown at different time instants t_i. As expected, agents move towards the left part of the area where the cells with the lowest d_w values are. At t_{10} the cell with center coordinates (10,10) has a low coverage level. However, due to the existence of the terminal constraints an agent can reach it within 4 time steps before (3) gets violated. A video of the simulations for Scenario 1 can be found in [38]. The average computational time of the online optimization problem over the simulation horizon of 100 steps is 115 sec. Observe that the computational time of the online problem is moderate due to the existence of the terminal constraints which most likely decrease the set of feasible solutions. The computational time of the closed trajectories is significantly higher verifying the results of Section 5.1 but it is performed offline, hence does not affect the complexity of the online algorithm.

Comparison with a greedy policy To further illustrate the efficacy of our method, we consider Scenario 1 and compare the results of the proposed RHS to those obtained by a greedy algorithm, i.e., a method that plans the poses of the agents only for the next time step t_i. Motivated by [25,26], where agents move towards the least covered areas, we solve (10) for $N = 1$ without considering (9c), i.e., the constraint ensuring a lower coverage level $\underline{Z}$. As mentioned in Section 3, this problem aims at maximizing the total coverage level of the area while discouraging agents from covering recently visited cells. As expected, the computational time of the problem is significantly lower than the RHS. However, after 12 time steps the coverage level of the cell with center coordinates (6,14), i.e., $w = 20$, becomes equal to 19.3341. Hence, it drops below the desired lower bound $\underline{Z} = 20$. In Figure (5) the coverage level of another cell, namely the one with center coordinates (4,10) is shown. When the greedy algorithm is considered, the cell is visited only at time steps t_i with $i = 3, 23$ while for $i \geq 40$ its coverage level is less than $\underline{Z}$ and $\mathbf{z}_{t_i}(16) \to 0$ as $i \to 100$. When (9c) is considered in the greedy algorithm, (10) becomes infeasible at t_i with $i = 13$ as no agent performing an admissible pose transition is able to cover $w = 20$ before its coverage level drops below $\underline{Z}$. On the other hand, as shown in Figure (5) and discussed earlier in this Section, the proposed RHS scheme ensures that the coverage level of all cells is lower bounded by $\underline{Z}$ and remains feasible for all t_i with $i \geq 0$ at the cost of increased computational complexity.

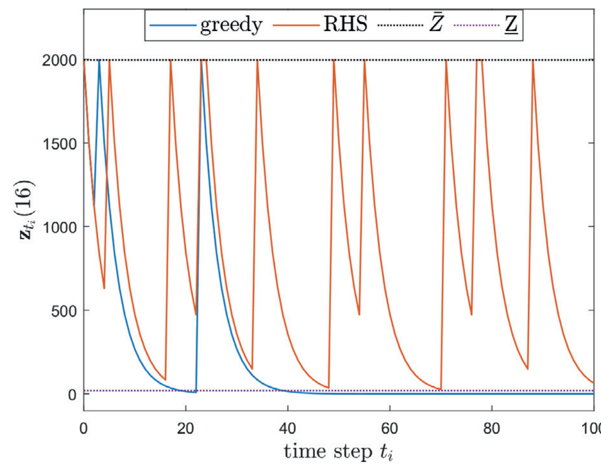


Fig. 5. Coverage Level of the cell with center coordinates (14,10) as a function of t_i, obtained by a greedy algorithm and the proposed RHS scheme.

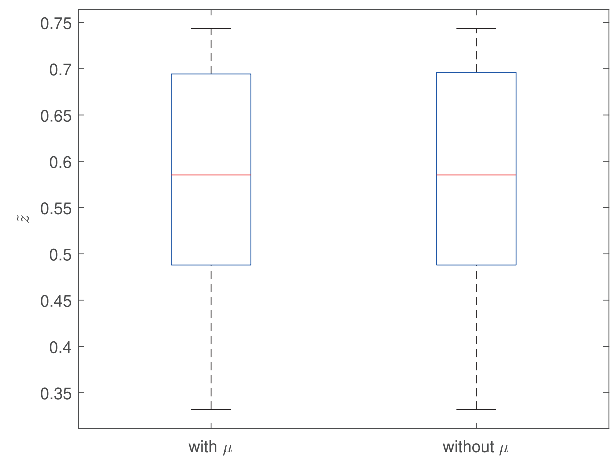


Fig. 6. Average normalized Coverage Level with respect to the simulation horizon for the problem with and without μ variables.

Comparison with no coverage penalization at subsequent times Next, we study the effect of the objective function in the trajectory design when the proposed RHS scheme is considered. More specifically, we consider solving (18) without considering the μ variables and corresponding constraints (5a)-(5b). As a result, the new objective function becomes $J''(\tilde{\mathbf{x}}) = \begin{bmatrix}\mathbf{0}_{\tilde{n}_b} & \mathbf{1}_{\tilde{n}_z} & \mathbf{0}_{\tilde{n}_\mu} & \mathbf{0}_{\tilde{n}_\kappa}\end{bmatrix}\tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} = \begin{bmatrix}\mathbf{b}^T & \mathbf{z}^T & \boldsymbol{\kappa}^T\end{bmatrix}^T$. For the problem without the μ variables, the optimal $\bar{Z}$ value is found using the proposed two-step method equal to 1995.5 as in (18). In Figure (6), the normalized coverage level of the cells, averaged over the simulation horizon of 100 time steps is shown, when the RHS scheme with and without the μ variables and corresponding constraints is solved. Specifically, we compute $\tilde{z}$ as follows:

$$\tilde{z} = \frac{1}{100\bar{Z}} \sum_{k=1}^{100} \mathbf{z}_k(w), \qquad \forall w \in I$$

Although, there are cells whose coverage level might be higher when the constraints and μ variables are omitted, the average coverage level in both cases is equal to 0.5822. While the average coverage level remains the same, the computational time of the problem when no μ constraints are considered increases to 143.61 sec on average (as opposed to 129.8 sec of (18)). In both problems the agents have avoided interchanging cells with only 1 case of "flipping" over 100 steps. Finally, the minimum time between two consecutive visits (excluding the cases when an agent stays at place after coverage) is on average 20% of the worst-case upper bound of Proposition 1. Based on the above, we can conclude that for the given scenario the presence of the $\boldsymbol{\mu}$ variables is beneficial both in terms of coverage performance and computational complexity.

6. Conclusions

In this work we introduced an MILP problem for planning the trajectories of agents performing a persistent coverage task in a grid environment. In this task the goal of the agents is to maintain a predefined lower bound on the coverage level of each cell when the coverage level dynamics are known. Due to the special design of the problem a lower bound on the frequency of visitation of each cell is also guaranteed. In addition, a modified version of the problem is presented that is found to be recursively feasible when solved in a receding horizon scheme. The key difference between the aforementioned problems is an extra set of time-varying terminal constraints added to the latter problem. These constraints force agents to terminate their plans at closed trajectories at the end of the planning horizon in order to guarantee recursive feasibility. We proposed a two-step method for the construction of these closed trajectories such that the coverage level lower bound constraint is always satisfied when the trajectories are repeatedly followed.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to thank Mernout Burger from TBA Netherlands for the interesting discussions and the insightful comments on the formulation of the Mixed Integer Problem.

References

[1] N. Agmon, N. Hazon, G. Kaminka, Constructing spanning trees for efficient multi-robot coverage, in: Proceedings of the International Conference on Robotics and Automation, 2006, pp. 1698–1703.
[2] I. Al-Taharwa, A. Sheta, M. Al-Weshah, A mobile robot path planning using genetic algorithm in static environment, J. Comput. Sci. 4 (4) (2008) 341–344.
[3] C. Azevedo, B. Lacerda, N. Hawes, P. Lima, Steady-state policy synthesis for verifiable control, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 5653–5661.
[4] C. Azevedo, B. Lacerda, N. Hawes, P. Lima, Long-run multi-robot planning under uncertain action durations for persistent tasks, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 4323–4328.
[5] R. Bobiti, M. Lazar, On the computation of lyapunov functions for discrete-time nonlinear systems, in: 18th International Conference on System Theory, Control and Computing, 2014, pp. 93–98.
[6] F. Bullo, J. Cortés, S. Martínez, Distributed Control of Robotic Networks, Applied Mathematics Series, Princeton University Press, 2009.
[7] M. Burger, Z. Su, B.D. Schutter, A node current-based 2-index formulation for the fixed-destination multi-depot travelling salesman problem, Eur. J. Operation. Res. 265 (2) (2018) 463–477.
[8] D. Casbeer, D. Kingston, R. Beard, T. McLain, Cooperative forest fire surveillance using a team of small unmanned air vehicles, Int. J. Syst. Sci. 37 (6) (2006) 351–360.
[9] H. Chen, F. Allgöwer, A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability, Automatica 34 (10) (1998) 1205–1217.
[10] C.H. Chen, K.T. Song, Complete coverage motion control of a cleaning robot using infrared sensors, in: IEEE International Conference on Mechatronics, 2005, pp. 543–548.
[11] H. Choset, Coverage for robotics-a survey of recent results, Annal. Math. Intell. 31 (2001) 113–126.
[12] C. Franco, D. Paesa, G. Lopez-Nicolas, C. Sagüés, S. Llorente, Hierarchical strategy for dynamic coverage, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 5341–5346.
[13] Y. Gabriely, E. Rimon, Spiral-STC: an on-line coverage algorithm of grid environments by a mobile robot, in: IEEE International Conference on Robotics and Automation, 1, 2002, pp. 954–960.
[14] I. Hussein, D. Stipanović, Effective Coverage Control using Dynamic Sensor Networks with Flocking and Guaranteed Collision Avoidance, in: American Control Conference, 2007, pp. 3420–3425.
[15] R. Kala, A. Shukla, R. Tiwari, S. Rungta, R.R. Janghel, Mobile robot navigation control in moving obstacle environment using genetic algorithm, artificial neural networks and a* algorithm, in: WRI World Congress on Computer Science and Information Engineering, 2009, pp. 705–713.
[16] S.M. LaValle, Planning Algorithms, Cambridge University Press, New York, 2006.
[17] M. Lazar, V. Spinu, Finite-step terminal ingredients for stabilizing model predictive control, IFAC PapersOnLine 48 (23) (2015) 9–15.
[18] D. Limon, I. Alvarado, T. Alamo, E.F. Camacho, Robust tube-based mpc for tracking of constrained linear systems with additive disturbances, J. Process Control 20 (3) (2010) 248–260.
[19] D. Limon, M. Pereira, D.M. noz de la Peña, T. Alamo, C.N. Jones, M.N. Zeilinger, MPC For tracking periodic references, IEEE Trans. Automat. Control 61 (4) (2016) 1123–1128.
[20] X. Lin, C. Cassandras, An optimal control approach to the multi-Agent persistent monitoring problem in two-Dimensional spaces, IEEE Trans. Automat. Control 60 (6) (2015) 1659–1664.
[21] M. Lorenzen, F. Dabbene, R. Tempo, F. Allöwer, Constraint-tightening and stability in stochastic model predictive control, IEEE Trans. Automat. Control 62 (7) (2017) 3165–3177.
[22] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: stability and optimality, Automatica 36 (6) (2000) 789–814.
[23] I. Maza, A. Ollero, Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms, in: R. Alami, R. Chatila, H. Asama (Eds.), Distributed Autonomous Robotic Systems, 6, Springer Tokyo, 2007, pp. 221–230.
[24] N. Nigam, The multiple unmanned air vehicle persistent surveillance problem: a review, Machines 2 (1) (2014) 13–72.
[25] N. Nigam, S. Bieniawski, I. Kroo, J. Vian, Control of multiple UAVs for persistent surveillance: algorithm and flight test results, IEEE Trans. Control Syst. Technol. 20 (5) (2012) 1236–1251.
[26] J.M. Palacios-Gasós, E. Montijano, C. Sagüés, S. Llorente, Distributed coverage estimation and control for multirobot persistent tasks, IEEE Trans. Robot. 32 (6) (2016) 1444–1460.
[27] J. Palacios-Gasós, E. Montijano, C. Sag'és, S. Llorente, Cooperative periodic coverage with collision avoidance, IEEE Trans. Control Syst. Technol. 27 (4) (2019) 1411–1422.
[28] J. Palacios-Gasós, Z. Talebpour, E. Montijano, C. Sagüés, A. Martinoli, Optimal path planning and coverage control for multi-robot persistent coverage in environments with obstacles, in: IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 1321–1327.
[29] D. Panagou, D. Stipanović, P. Voulgaris, Distributed dynamic coverage and avoidance control under anisotropic sensing, IEEE Trans. Control Netw. Syst. 4 (4) (2016) 850–862.
[30] F. Pasqualetti, A. Franchi, F. Bullo, On cooperative patrolling: optimal trajectories, complexity analysis, and approximation algorithms, IEEE Trans. Robot. 28 (3) (2012) 592–606.
[31] I. Rekleitis, A. New, E. Rankin, H. Choset, Efficient boustrophedon multi-Robot coverage: an algorithmic approach, Annal. Math. Artif. Intell. 52 (2–4) (2008) 109–142.
[32] S. Smith, D. Rus, Multi-robot monitoring in dynamic environments with guaranteed currency of observations, in: 49th IEEE Conference on Decision and Control (CDC), 2010, pp. 514–521.
[33] S. Smith, M. Schwager, D. Rus, Persistent robotic tasks: monitoring and sweeping in changing environments, IEEE Trans. Robot. 28 (2) (2012) 410–426.
[34] R. Smith, M. Schwager, S. Smith, D. Rus, G. Sukhatme, Persistent ocean monitoring with underwater gliders: Towards accurate reconstruction of dynamic ocean processes, in: IEEE International Conference on Robotics and Automation, 2011, pp. 1517–1524.
[35] C. Song, L. Liu, G. Feng, S. Xu, Optimal control for multi-agent persistent monitoring, Automatica 50 (6) (2014) 1663–1668.
[36] G. Sun, R. Zhou, B. Di, Z. Dong, Y. Wang, A novel cooperative path planning for multi-robot persistent coverage with obstacles and coverage period constraints, Sensors(Basel) 19 (9) (2019).
[37] J. Tu, S.X. Yang, Genetic algorithm based path planning for a mobile robot, in: IEEE International Conference on Robotics and Automation, 2003, pp. 1221–1226.
[38] Video for scenario 1, 2020, (https://youtu.be/hXBzEJZ-wE8). Online; accessed 13 July 2020.
[39] N. Zagradjanin, D. Pamucar, K. Jovanovic, Cloud-based multi-robot path planning in complex and crowded environment with multi-criteria decision making using full consistency method, Symmetry 11 (10) (2019) 1241.
[40] X. Zheng, S. Jain, S. Koenig, D. Kempe, Multi-robot forest coverage, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, pp. 3852–3857.

[41] N. Zhou, C.G. Cassandras, X. Yu, S.B. Andersson, Optimal event-Driven multi-Agent persistent monitoring with graph-Limited mobility, IFAC-PapersOnLine 50 (1) (2017) 2181–2186.
[42] N. Zhou, C.G. Cassandras, X. Yu, S.B. Andersson, Decentralized event-driven algorithms for multi-agent persistent monitoring tasks, in: IEEE 56th Annual Conference on Decision and Control, 2017, pp. 4064–4069.
[43] N. Zhou, X. Yu, S.B. Andersson, C.G. Cassandras, Optimal event-Driven multiagent persistent monitoring of a finite set of data sources, IEEE Trans. Automat. Control 63 (12) (2018) 4204–4217.