Cooperative Periodic Coverage with Collision Avoidance

José Manuel Palacios-Gasós, *Student Member, IEEE,* Eduardo Montijano, *Member, IEEE,* Carlos Sagüés, *Senior Member, IEEE,* and Sergio Llorente

***Abstract*—In this paper we propose a periodic solution to the problem of persistently covering a finite set of interest points with a group of autonomous mobile agents. These agents visit periodically the points and spend some time carrying out the coverage task, which we call coverage time. Since this periodic persistent coverage problem is NP-hard, we split it into three subproblems to counteract its complexity. In the first place, we plan individual closed paths for the agents to cover all the points. Second, we formulate a quadratically constrained linear program to find the optimal coverage times and actions that satisfy the coverage objective. Finally, we join together the individual plans of the agents in a periodic team plan by obtaining a schedule that guarantees collision avoidance. To this end, we solve a mixed integer linear program that minimizes the time in which two or more agents move at the same time. Eventually, we apply the proposed solution to an induction hob with mobile inductors for a domestic heating application and show its performance with experiments on a real prototype.**

I. Introduction

In recent years, the development of autonomous mobile agents has experienced an important growth and has provided them the capabilities to carry out a great variety of tasks while being affordable. This has motivated an increasing research interest in multi-agent systems that are capable of accomplishing tasks as a team that a single agent cannot [1], [2]. A particularly interesting problem in this context is that of persistent coverage, which finds applications in many fields such as cleaning [3], environmental monitoring [4], [5] or aerial delivery [6]. The objective of persistent coverage is to maintain covered over time a dynamic environment in which the coverage level persistently deteriorates. Coverage can be seen as a measure of how dirty is the environment in a cleaning application, how hot in a heating one or how well observed in monitoring. As time goes by, the environment gets dirty, cools down or the accuracy of the observations degrades, respectively. The agents must spend some time at each point of the environment and keep moving to maintain the coverage of the entire environment at a desired level. This is the main difference with more traditional coverage approaches [7].

A. Related Work

Different types of solutions, which can be indefinitely applied [8], have been proposed. The first type are controller-based approaches as [9], where a gradient descent method together with an assignment of objectives is proposed. In [10] a control policy decides to which cell move next to minimize the maximum time between visits to all cells and Branch and Bound is used in [11] to find the optimal control inputs of the agents for a finite prediction horizon.

The second kind of proposals intend to plan closed paths that the agents repeatedly follow. The objective is to minimize some coverage metric while periodically visiting all the points of the environment. To minimize the estimation uncertainty, a parametric optimization is formulated in [12], Rapidly-exploring Random Trees are used in [13] and Rapidly-exploring Random Cycles in [14]. In [15], they aim to minimize the maximum time that points remain unvisited by planning paths through the vertices of a graph and [16] guarantees that full awareness is eventually reached in some interest points. In these proposals, the time that each agent needs to perform the coverage task at each point is not considered and the movements have priority over the coverage of particular points.

In discrete environments, where only a finite set of points requires coverage, approaches are related to Task Assignment [17] and Vehicle Routing Problems (VRP) as addressed in operational research [18]. Since VRP are known to be NP-hard, three different types of solutions appear in the literature: based on heuristics, formulated as Mixed Integer Programs (MIP), although they suffer from complexity for large problems, or a combination of both. In [19] a Mixed-Integer Linear Program (MILP) and a genetic algorithm are compared for the scheduling of mission trajectories. To find a closed path to visit all the points of the environment with a single UAV, heuristic solutions are presented with refueling depots [20] and with revisit constraints [21]. In [22], they seek the optimal routing strategy using a linear MIP that allows a fleet of vehicles to periodically visit a set of targets while trying to minimize the energy consumption. Similarly, in [6] they maximize the frequency of task completion allowing online calculation of task costs to avoid energy depletion. These approaches assume that only one agent covers each point. In [23] a utility metric is maximized through a quadratic MIP to find the best tours for the agents when correlation between the points of interest is considered.

Most of the aforementioned works assume that the times

J.M. Palacios-Gasós, E. Montijano and C. Sagüés are with Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza, Spain. {jmpala, emonti, csagues}@unizar.es

S. Llorente is with Research and Development Department, Induction Technology, Product Division Cookers, BSH Home Appliances Group, 50016, Zaragoza, Spain. sergio.llorente@bshg.com

This work was partially supported by projects RTC-2014-1847-6 of Retos-Colaboración, DPI2015-69376-R from Ministerio de Economía y Competitividad/Unión Europea, DGA T04-FSE, CUD2016-17 and DGA Scholarship C076/2014, partially funded by European Social Fund.

required to complete the coverage task at each point are known or do not consider them. The latter is acceptable in monitoring or surveillance applications [24], since the information gathering can be considered instantaneous, but the same does not hold in problems such as heating or watering, where the coverage action of the agent requires some time. Moreover, the possibility of collisions between agents along their paths is usually not addressed.

The problem of calculating the times that the agents have to spend at each point to satisfy the coverage objective has not been deeply studied in the literature. In [16] and [25] velocity controllers are calculated to spend more time covering the points where the environment changes quickly. In our previous work [26], on which this work builds, the optimal coverage times for each pair agent-point are calculated when the actions that the agents apply are fixed. Without a coverage objective to accomplish, in [27] a single agent obtains a reward depending on the time that it covers each point and a path is calculated to obtain the maximum reward with a limited amount of fuel.

The problem of collision avoidance is considered in other multi-agent works by scheduling the agent trajectories. This is usually solved by introducing time delays at the beginning of the paths [28]. In [29] they also assign goals to interchangeable agents to minimize the maximum cost over all trajectories.

B. *Contributions*

In this paper we propose a solution to persistently cover a finite set of points with a team of agents. These agents must spend some time at each point to improve its coverage. We find a periodic strategy for the team, composed of individual periodic paths, coverage times and actions, that guarantees the satisfaction of the coverage objective for all the points, optimizes the actions of the agents and avoids collisions between them.

Our main contribution is a complete solution to the problem using a divide-and-conquer strategy. We separate the problem into three subproblems: (i) planning individual, closed paths for the agents that cover all the points of the environment; (ii) calculating the coverage time and coverage action of each agent at each point of its path to satisfy the coverage objective; and (iii) scheduling the start of the agent paths to obtain a team plan in which collisions are avoided.

As mentioned before, the complete problem is hard to solve at once because the computational complexity grows exponentially with the number of agents and points. The divide-and-conquer methodology counteracts this growth and allows us to find the optimal solution for each subproblem independently. Moreover, although the final solution to the entire problem may not be the optimal, this is a conceptually simple and piecewise optimal way of calculating a solution to the problem. In fact, this is the first attempt to calculate simultaneously the paths and the coverage times and actions, that allows the collaboration of several agents at the same point and guarantees collision avoidance.

In the paper we do not elaborate on subproblem (i), since it reduces to solve an instance of the Traveling Salesman Problem (TSP) for which there are state-of-the-art solutions. Regarding subproblems (ii) and (iii), our contributions in this work are:

- A generalization of our previous work [26] that includes the coverage actions with the coverage times in subproblem (ii). It allows us to optimize the coverage satisfying the objective and minimizing the instantaneous actions of the agents.
- A procedure to schedule the agent paths that provides collision avoidance guarantees and minimizes simultaneous movements of the agents.
- An application of the solution to the problem of domestic induction heating with mobile inductors, that is an open problem which offers a significant opportunity to increase the flexibility of domestic hobs at a moderate increase of cost.
- Real experiments with a prototype for this application.

The remainder of the paper is structured as follows. In Section II we introduce the problem formulation and an overview of the proposed solution. We calculate the optimal coverage times and actions in Section III and face the team plan scheduling problem in Section IV. Finally, we present simulation results in Section V, the application to domestic induction heating with experimental results in Section VI and conclusions in Section VII.

II. Problem Formulation and Solution Overview

Let $\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_Q\}$ be a finite set of Q points of interest that must be covered. At each point, we define the coverage level with a scalar field, $Z(\mathbf{q}, t) \geq 0$. The objective is to maintain a desired level $Z^*(\mathbf{q}, t) \geq 0$ by providing a certain coverage action $P^*(\mathbf{q}, t) \geq 0$ over time, not necessarily the same for all the points. This is achieved using a team of $I \in \mathbb{N}$ mobile agents $\mathcal{I} = \{i_1, \dots, i_I\}$ of radius r_i, $i \in \mathcal{I}$. They are capable of increasing the level at the points in which they are located according to a *production function*, $P_i(\mathbf{q}, t)$:

$$P_i(\mathbf{q}, t) = \begin{cases} 0, & \text{if } \mathbf{p}_i(t) \neq \mathbf{q}, \\ \rho_i(\mathbf{q}), & \text{if } \mathbf{p}_i(t) = \mathbf{q}, \end{cases}$$

where $\mathbf{p}_i(t)$ is the position of the agent at time $t \geq 0$ and $0 \leq \rho_i(\mathbf{q}) \leq \rho_i^{\max}(\mathbf{q})$ is the *coverage action* of the agent, which can also be controlled. The maximum production $\rho_i^{\max}(\mathbf{q})$ can be different for each agent-point pair.

Each agent may not be capable of reaching all the points due to physical constraints. The subset $\mathcal{Q}_i \subseteq \mathcal{Q}$ represents the *reachable set* of points for agent i, with the team satisfying $\cup_{i \in \mathcal{I}} \mathcal{Q}_i = \mathcal{Q}$. Reciprocally, each point $\mathbf{q} \in \mathcal{Q}$ can be covered by a set of visiting agents, $\mathcal{I}_{\mathbf{q}} \subseteq \mathcal{I}$.

In this context, the coverage provided at each point can be calculated as

$$Z(\mathbf{q}, t) = \int_0^t \sum\nolimits_{i \in \mathcal{I}_{\mathbf{q}}} P_i(\mathbf{q}, \tau) d\tau, \tag{1}$$

assuming $Z(\mathbf{q}, 0) = 0$ as initial condition, and the desired coverage as

$$Z^*(\mathbf{q}, t) = \int_0^t P^*(\mathbf{q}, \tau) d\tau. \tag{2}$$

A solution that maintains all the time $Z(\mathbf{q},\tau) = Z^*(\mathbf{q},\tau)$ requires at least as many agents as points and does not exist if the number of agents is lower than the number of points. Since this is usually the case, an optimization metric is defined to try to maintain $Z(\mathbf{q},\tau)$ as close as possible to $Z^*(\mathbf{q},\tau)$ over time. Some optimality criteria could be to minimize over time the quadratic difference between the required and the provided coverage, i.e., minimize $\int_0^t \sum_{\mathbf{q}\in\mathcal{Q}} \big(Z^*(\mathbf{q},\tau) - Z(\mathbf{q},\tau)\big)^2 d\tau$, or to minimize the maximum difference between $Z(\mathbf{q},t)$ and $Z^*(\mathbf{q},t)$ over time, i.e., minimize $\max\big(Z^*(\mathbf{q},t) - Z(\mathbf{q},t)\big)$. However, trying to minimize globally a function associated with these metrics results in a problem that has been proven to be NP-hard. For these reason, we seek to guarantee that the desired coverage is provided periodically with period T. A periodic approach, though inherently suboptimal with respect to such metrics and still NP-hard, allows us to guarantee that all the points receive on average the coverage that they require, with a repetitive strategy that can be calculated in advance. In this scenario, the periodic objective requires that (i) at some time $t \le T$ the coverage reaches $Z(\mathbf{q},t) = Z^*(\mathbf{q},t)$, and (ii) from that time on, $Z(\mathbf{q},t+kT) = Z^*(\mathbf{q},t+kT)$, $\forall\, k \in \mathbb{Z}$. Equivalently,

$$Z(\mathbf{q},t+kT) - Z\big(\mathbf{q},t+(k-1)\,T\big) = \\ Z^*(\mathbf{q},t+kT) - Z^*\big(\mathbf{q},t+(k-1)\,T\big).$$

We can assume that $P^*(\mathbf{q}) \equiv P^*(\mathbf{q},t)$ is constant over time, or at least periodic, since the rate of change of $P^*(\mathbf{q})$ is usually much bigger than the period. Thus, the problem can be considered stationary between changes. Then, the right term is equal to $P^*(\mathbf{q},\tau)\,T$ according to (2) and introducing (1) the objective becomes

$$\int_{t+(k-1)T}^{t+kT} \sum\nolimits_{i\in\mathcal{I}_\mathbf{q}} P_i(\mathbf{q},\tau) d\tau = P^*(\mathbf{q})\,T.$$

In order to satisfy this objective, the agents follow a periodic path, Γ_i, between reachable points, i.e., an ordered subset of $\mathcal{Q}_i$. Along these paths, the agents spend some time, namely the *coverage time* $\theta_{i,\Gamma_i(j)}$, covering each point. These times are normalized by the period, i.e., they represent the fraction of the period that a robot is covering a point and allow us to formulate the problem independently of the actual value of the period. With this normalization, the periodic coverage objective for each point $\mathbf{q}$ of the environment can be stated as

$$\sum\nolimits_{i\in\mathcal{I}_\mathbf{q}} \rho_{i,\Gamma_i(j)}\,\theta_{i,\Gamma_i(j)} = P^*(\mathbf{q}).$$

With a little abuse of notation we refer to $\mathbf{q}$ as the j-th point in the path of agent i, i.e., $\Gamma_i(j) \equiv \mathbf{q}$, although j may be different for each agent i. In the following we refer to it only with subindex j for simplicity. Thus, the objective of the problem is to find the paths Γ_i, coverage times $\theta_{i,j}$ and productions $\rho_{i,j}$ that satisfy

$$\sum\nolimits_{i\in\mathcal{I}_\mathbf{q}} \rho_{i,j}\,\theta_{i,j} = P^*(\mathbf{q}) \tag{3}$$

and do not lead to a collision.

Each point is visited only once per period by the agent and each visit has associated two additional times: the *arrival time*, $a_{i,j}$, the instant in which the agent arrives and starts covering a point, and the *departure time*, $d_{i,j}$, the instant in which the agent stops covering the point and leaves towards the next one. The total (normalized) moving time, i.e., the time that agent i would need to traverse Γ_i without making any stops, is θ_i^m. Note that the periodic plans can be repeated as long as all $\mathcal{Q}_i$, $\mathcal{I}_\mathbf{q}$ and $P^*(\mathbf{q})$ remain constant. Also note that they are independent of the value of the period and that such value can be chosen differently depending on the application.

The solution that we propose to this problem follows a divide-and-conquer strategy, as shown in Alg. 1. We separate the problem into three subproblems (Steps 1, 2 and 4) to reduce its complexity and make it tractable. Even though it does not guarantee global optimality, this allows us to solve efficiently the three subproblems and guarantee periodic coverage. In fact, the second and third subproblems can be solved in an optimal manner and for the first one, that has been deeply studied in the literature, there are many efficient methods with guaranteed closeness to the optimum.

Algorithm 1 Solution Overview

```
1: Plan initial paths.
2: Calculate optimal coverage times and actions.
3: Shorten paths and recalculate times and actions.
4: Schedule the path starts to avoid collisions.
```

In the first step, we find the initial path for each robot. This path is the shortest, closed one that visits all the reachable points of the robot, $\mathcal{Q}_i$. To do so, we solve a simple Traveling Salesman Problem (TSP) [30] using an order-first split-second approach [31]. We do not elaborate further on this step since many solutions have been proposed and analyzed in the literature. For the second step, we pose a quadratically constrained linear program with the restrictions of satisfying the periodic coverage objective from (3). The algorithm to shorten the paths can be seen in our previous work [26]. In the last step, we calculate a schedule that allows us to include the individual periodic plans of the agents in a periodic team plan in which collision avoidance is guaranteed. We formulate this scheduling as a MILP to minimize the time in which two or more agents are moving at the same time while satisfying collision avoidance restrictions. The problem could be limited to satisfy them but it is worth to optimize another criteria at the same time. Actually, the restrictions might be included in the problem of the times and production, but this would exponentially increase the complexity of the problem and would still not guarantee the global optimum to be found.

In the following sections we explain in detail the two main steps of our solution, namely Steps 2 and 4.

III. Optimal Times and Productions

The periodic coverage objective requires the calculation of the coverage times and coverage actions to be guaranteed, with the periodic paths that the agents travel. The problem of calculating these times and productions reduces to find the times $\theta_{i,j}$ and productions $\rho_{i,j}$ that comply with (3) and with the periodicity of the system. To this end, we consider a cost function on the times and the productions, $f(\theta_{i,j}, \rho_{i,j})$, and the resulting quadratically constrained program is

$$\begin{aligned} \underset{\theta_{i,j},\rho_{i,j}}{\text{minimize}} \quad & f(\theta_{i,j}, \rho_{i,j}) \\ \text{subject to} \quad & \sum_{i \in \mathcal{I}_{\mathbf{q}}} \rho_{i,j}\, \theta_{i,j} = P^*(\mathbf{q}), \qquad \forall\, \mathbf{q} \in \mathcal{Q}, \qquad \text{(4a)} \\ & \sum_{\mathbf{q} \in \mathcal{Q}_i} \theta_{i,j} \leq 1 - \theta_i^m, \qquad \forall\, i \in \mathcal{I}, \qquad \text{(4b)} \\ & \sum_{i \in \mathcal{I}_{\mathbf{q}}} \theta_{i,j} \leq 1, \qquad \forall\, \mathbf{q} \in \mathcal{Q}. \qquad \text{(4c)} \end{aligned}$$

Eq. (4a) is the quadratic restriction on the coverage objective. of a linear program that has already been solved distributedly [32] for multi-robot systems [33]. The set of equations (4b) imposes that for each agent the time spent covering the points plus the time to move along the path must be lower than the period, and the set (4c) represents that each point cannot be covered for more time than the period. This restriction is only needed in the case that it is not allowed that two agents cover the same point at the same time because otherwise it would be impossible to avoid collisions between agents. The selection of the cost function is entirely dependent on the performance expected from the particular application. For this reason, in Section V we explore and analyze different linear cost functions that are appropriate for many applications.

This optimization problem is a generalization of our previous approach [26] and includes the productions of the agents at each point as variables of the problem. Nevertheless, the sufficient conditions on the existence of solution presented in [26] are still valid. These conditions allow us to know a priori if the agents are capable of satisfying the coverage objective.

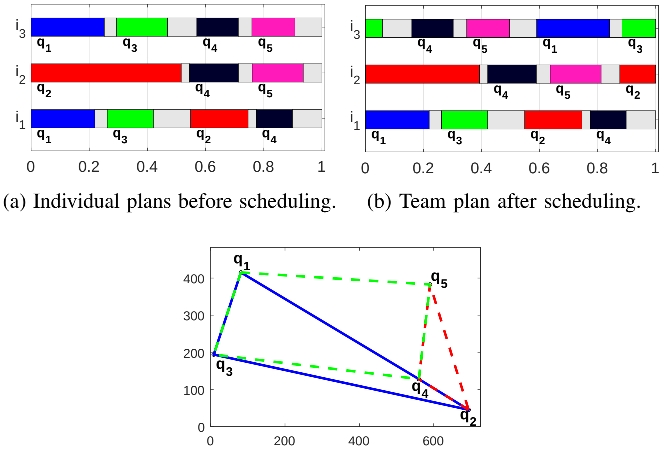
A particularity of the solutions obtained for (4) is that some of the times $\theta_{i,j}$ may be equal to zero. This implies that agent i is not required to cover point j. Thus, shorter paths can be followed only through points with coverage times greater than zero, reducing the moving times θ_i^m and, therefore, the total time required for the coverage of the environment.

In our previous work [26] we proposed an iterative algorithm to shorten the paths based on this property that links the first two subproblems. At each iteration, the algorithm optimizes the agent paths using the results of the previous optimal coverage times and productions. Then, the times and productions are optimized again for the new refined paths. This makes the combination of both to iteratively improve the global solution achievable with this periodic approach.

IV. Team Plan Scheduling

We devote this section to the calculation of a scheduling for the start of the periodic paths of the agents. It avoids collisions while covering a point and while moving between points. The intuitive idea of the scheduling is to shift the individual plans of the agents in time to obtain a collision-free plan for the entire team, that is, to find a time, $0 \leq \varphi_i < 1$, for each agent $i \in \{2, \dots, I\}$ such that the execution of its periodic plan shifted by this time leads to no collision with the others. We refer the initial times to the beginning of the path of agent $i = 1$, i.e., we fix $\varphi_1 = 0$, and the paths remain the same through the scheduling.

In Fig. 1 we show an example of scheduling for $I = 3$ agents of size $r_i = 5$ and $Q = 5$ points. Fig. 1a shows the individual plan of each agent, which includes the path (order in which points are visited) and the coverage times (width of the colored rectangles). The beginning and end of the colored rectangles correspond to the arrival and departure times, $a_{i,j}$ and $d_{i,j}$, respectively, and the gray rectangles represents the time needed to move from one point to the following. It can be seen that some coverages of the same point are overlapped in time and, therefore, the execution of these individual plans leads to collisions. For instance, it happens when i_1 and i_3 try to cover $\mathbf{q}_1$ or $\mathbf{q}_3$, or when i_2 and i_3 try to cover $\mathbf{q}_4$ or $\mathbf{q}_5$. In Fig. 1b, where the team plan after the scheduling is depicted, it can be seen that the individual plans of agents i_2 and i_3 have been shifted an 88% and a 59% of the period respectively and that in the resulting plan there are no overlaps between coverages of the same point. Fig. 1c shows the locations of the points and the paths followed by the agents. It can be seen that no collisions occur after the scheduling.



(a) Individual plans before scheduling.

(b) Team plan after scheduling.

(c) Agent paths.

Fig. 1. Example of scheduling. (a)-(b) Each row represents the plan of an agent. The points $\mathbf{q}_i$ that each agent covers are represented in different colors. The coverage times are depicted by the width of the colored rectangles and the gray rectangles represent the time needed to move between points. (c) Paths followed by the agents i_1, i_2 and i_3 in blue, red and green, respectively. The radius of the agents is $r_i = 5$ units.

The calculation of the optimal schedule is done by solving a MILP with constraints. In the following subsections we develop the formulation, the restrictions and the cost function, and finally pose the problem and discuss its suitability.

A. Transformation of Individual Times to Team Plan

The previous step to calculate the schedule that avoids collisions is to refer the individual times of the agents to the team plan. The arrival times can be expressed as follows:

$$\begin{aligned} A_{i,j} &= \varphi_i + a_{i,j}, \qquad && \text{if } \varphi_i + a_{i,j} \leq 1, \qquad \text{(5a)} \\ A_{i,j} &= \varphi_i + a_{i,j} - 1, \qquad && \text{if } \varphi_i + a_{i,j} > 1, \qquad \text{(5b)} \end{aligned}$$

for all $j \in \{2, \dots, |\Gamma_i|\}$, $i \in \{2, \dots, I\}$. Note that (5) does not apply for $j = 1$ since $a_{i,1} = 0$ and $A_{i,1} = \varphi_i$, and recall that for agent $i = 1$ we set $\varphi_1 = 0$ and, therefore, its corresponding times do not need transformation.

To be able to include these times in the MILP formulation, we introduce the binary variables, $c^a_{i,j}$, such that, $c^a_{i,j} = 0$ if $\varphi_i + a_{i,j} \leq 1$ and $c^a_{i,j} = 1$, otherwise. These variables represent if the arrival times in the team plan are bigger than 1 and they have to be shifted to the left side of the team plan by substracting 1, e.g., the arrival time of i_2 to $\mathbf{q}_4$ in Fig. 1b.

Thus, we can express the arrival times as

$$A_{i,j} = \varphi_i + a_{i,j} - c^a_{i,j}$$

subject to

$$\varphi_i + a_{i,j} \leq 1 + R\, c^a_{i,j}, \tag{6a}$$
$$-(\varphi_i + a_{i,j}) \leq -1 + R\,(1 - c^a_{i,j}), \tag{6b}$$

obtained with the big number method [34]. This method activates or deactivates the constraints depending on the value of the binary variable with R being a sufficiently big number. In this case, $c^a_{i,j} = 0$ if (5a) has to be used and $c^a_{i,j} = 1$ if (5b) is needed.

The same formulation can be developed for the departure times:

$$\begin{aligned} D_{i,j} &= \varphi_i + d_{i,j}, && \text{if } \varphi_i + d_{i,j} \leq 1, \\ D_{i,j} &= \varphi_i + d_{i,j} - 1, && \text{if } \varphi_i + d_{i,j} > 1, \end{aligned}$$

for all $j \in \{1, \ldots, |\Gamma_i|\}$, $i \in \{2, \ldots, I\}$. Introducing the binary variables $c^d_{i,j}$, we have

$$D_{i,j} = \varphi_i + d_{i,j} - c^d_{i,j}$$

subject to

$$\begin{aligned} \varphi_i + d_{i,j} &\leq 1 + R\, c^d_{i,j}, \\ -(\varphi_i + d_{i,j}) &\leq -1 + R\ (1 - c^d_{i,j}). \end{aligned}$$

Since the binary variables $c^a_{i,j}$ and $c^d_{i,j}$ are variables of the problem, we include the following restrictions on their values to guarantee that the order of the paths is followed:

$$\begin{aligned} c^a_{i,j} - c^d_{i,j} &\leq 0, \quad \forall\, j \in \{2, \ldots, |\Gamma_i|\}, \\ c^d_{i,j} - c^a_{i,j+1} &\leq 0, \quad \forall\, j \in \{1, \ldots, |\Gamma_i| - 1\}. \end{aligned}$$

B. Collision Avoidance During Coverage

A thorough consideration of collisions requires the inclusion of spatio-temporal restrictions in the problem, which implies a discretization of the environment and an analysis of what happens at each point [28]. Nevertheless, for large environments or with many agents or points to cover the problem becomes computationally unaffordable or even intractable. For this reason, we include the collision avoidance as planning constraints. Although it is overprotective, it is much lighter in terms of computational cost.

The first situation in which a collision may occur is when two or more agents try to cover the same point at the same time. From the planning perspective, this happens if the coverages of such point by different agents are overlapped in the team plan. We avoid this type of collisions with two groups of constraints. The first group is

$$A_{i_1,j_1} - A_{i_2,j_2} \leq R\,(1 - c^{cc}_{i_1.j_1,i_2,j_2}), \tag{7a}$$
$$A_{i_1,j_1} - D_{i_1,j_1} \leq R\,(1 - c^{cc}_{i_1.j_1,i_2,j_2}), \tag{7b}$$
$$D_{i_1,j_1} - A_{i_2,j_2} \leq -\varepsilon + R\,(1 - c^{cc}_{i_1.j_1,i_2,j_2}), \tag{7c}$$
$$\begin{aligned} D_{i_2,j_2} - A_{i_1,j_1} \leq& -\varepsilon + R\,(1 - c^{cc}_{i_1.j_1,i_2,j_2}) \\ &+ R(1 - c^d_{i_2,j_2} + c^a_{i_2,j_2}), \end{aligned} \tag{7d}$$

for all j_1, j_2 such that $\Gamma_{i_1}(j_1) = \Gamma_{i_2}(j_2)$, where the constant ε is the minimum separation between departures and arrivals to the same point. The first restriction, Eq. (7a), activates this group if $A_{i_1,j_1} \leq A_{i_2,j_2}$, that is, if the coverage of agent i_1 starts before the coverage of agent i_2. When this happens, the binary variable $c^{cc}_{i_1.j_1,i_2,j_2} = 1$. Since i_1 starts covering earlier, we have to assure that its coverage is not split by the end of the period, that is, its departure time is not translated to the first part of the period, as for the coverage of $\mathbf{q}_3$ by i_3 in Fig. 1b. Eq. (7b) guarantees this. In the third place, Eq. (7c) assures that the coverage of agent i_2 starts after the coverage of i_1 has ended, i.e., $D_{i_1,j_1} \leq A_{i_2,j_2}$. The last restriction guarantees that, if the coverage of i_2 has to be split, i.e., $c^d_{i_2,j_2} - c^a_{i_1,j_1} = 1$, it finishes before the coverage of i_1 starts.

The second group of restrictions is

$$A_{i_2,j_2} - A_{i_1,j_1} \leq R\, c^{cc}_{i_1.j_1,i_2,j_2}, \tag{8a}$$
$$A_{i_2,j_2} - D_{i_2,j_2} \leq R\, c^{cc}_{i_1.j_1,i_2,j_2}, \tag{8b}$$
$$D_{i_2,j_2} - A_{i_1,j_1} \leq -\varepsilon + R\, c^{cc}_{i_1.j_1,i_2,j_2}, \tag{8c}$$
$$\begin{aligned} D_{i_1,j_1} - A_{i_2,j_2} \leq& -\varepsilon + R\, c^{cc}_{i_1.j_1,i_2,j_2} \\ &+ R\,(1 - c^d_{i_1,j_1} + c^a_{i_1,j_1}). \end{aligned} \tag{8d}$$

Opposite to the first one, this second group is activated when $A_{i_2,j_2} \leq A_{i_1,j_1}$ or, equivalently, when $c^{cc}_{i_1.j_1,i_2,j_2} = 0$.

C. Collision Avoidance During Motion

The second situation in which a collision between a pair of agents may occur is when both of them are moving or one is moving and the other is covering a point. This can be prevented in the same way that the collisions during coverage, by avoiding the overlap of the movements and coverages that may cause a conflict. In fact, we propose the same restrictions as in (7)-(8) for the two cases: the movements of two agents, and a movement of an agent and a coverage of a point by another agent.

We define as $\delta_{i,j}$ and $\alpha_{i,j}$ the initial and final times of the movement between $\Gamma_i(j)$ and $\Gamma_i(j+1)$, respectively. They are defined opposite to the coverage times since $\delta_{i,j} = d_{i,j}$ is the departure of the movement and $\alpha_{i,j} = a_{i,j+1}, j \in \{1, \ldots, |\Gamma_i| - 1\}$, is the arrival of the movement. For $j = |\Gamma_i|$, we have $\alpha_{i,|\Gamma_i|} = 1$. Similarly to the coverage times, we can express the departure and arrival times of the movements in the team plan as a function of the times in the agent period and the binary variables $c^a_{i,j}$ and $c^d_{i,j}$ as follows:

$$\Delta_{i,j} = \varphi_i + \delta_{i,j} - c^d_{i,j}, \tag{9a}$$
$$\Lambda_{i,j} = \varphi_i + \alpha_{i,j} - c^a_{i,j+1}. \tag{9b}$$

For $j = |\Gamma_i|$, we define the binary variable c_i^a that represents if the final time of the last movement of each agent is greater than one or not. It requires the following restrictions:

$$\begin{aligned} \varphi_i + \alpha_{i,|\Gamma_i|} &\leq 1 + R\, c_i^a, \\ -(\varphi_i + \alpha_{i,|\Gamma_i|}) &\leq -1 + R\,(1 - c_i^a), \\ c^d_{i,|\Gamma_i|} - c_i^a &\leq 0. \end{aligned}$$

In order to decide if two movements can lead to a collision, we calculate the minimum distance between the two trajectories of the movements, $d^{mm}_{i_1.j_1,i_2,j_2}$ and determine that a collision is possible if such distance is lower than the sum of the sizes of the agents, i.e., $d^{mm}_{i_1.j_1,i_2,j_2} < r_{i_1} + r_{i_2}$. If the pair of movements may result into conflict, the following two sets of constraints are included in the problem. These sets are the same as (7)-(8), respectively, but for the departure and arrival times of the movements. The first set,

$$\Delta_{i_1,j_1} - \Delta_{i_2,j_2} \leq R\,(1 - c^{mm}_{i_1.j_1,i_2,j_2}), \tag{10a}$$
$$\Delta_{i_1,j_1} - \Lambda_{i_1,j_1} \leq R\,(1 - c^{mm}_{i_1.j_1,i_2,j_2}), \tag{10b}$$
$$\Lambda_{i_1,j_1} - \Delta_{i_2,j_2} \leq -\varepsilon + R\,(1 - c^{mm}_{i_1.j_1,i_2,j_2}), \tag{10c}$$
$$\begin{aligned} \Lambda_{i_2,j_2} - \Delta_{i_1,j_1} \leq{} & -\varepsilon + R\,(1 - c^{mm}_{i_1.j_1,i_2,j_2}) \\ & + R\,(1 - c^d_{i_2,j_2} + c^a_{i_2,j_2+1}), \end{aligned} \tag{10d}$$

is active when the movement of agent i_1 starts before the movement of i_2, i.e., $\Delta_{i_1,j_1} \leq \Delta_{i_2,j_2}$ in Eq. (10a). This is represented by the binary variable $c^{mm}_{i_1.j_1,i_2,j_2} = 1$. The second constraint represents that the movement of agent i_1 cannot be split by the end of the period; the third one, that the movement of i_2 must start after the movement of i_1 has ended; and the last one that, if the movement of i_2 is split by the end of the period, it must finish before i_1 starts moving.

Equivalently, the second set of constraints activates when the movement of agent i_2 starts before the movement of i_1:

$$\Delta_{i_2,j_2} - \Delta_{i_1,j_1} \leq R\, c^{mm}_{i_1.j_1,i_2,j_2}, \tag{11a}$$
$$\Delta_{i_2,j_2} - \Lambda_{i_2,j_2} \leq R\, c^{mm}_{i_1.j_1,i_2,j_2}, \tag{11b}$$
$$\Lambda_{i_2,j_2} - \Delta_{i_1,j_1} \leq -\varepsilon + R\, c^{mm}_{i_1.j_1,i_2,j_2}, \tag{11c}$$
$$\begin{aligned} \Lambda_{i_1,j_1} - \Delta_{i_2,j_2} \leq{} & -\varepsilon + R\, c^{mm}_{i_1.j_1,i_2,j_2} \\ & + R\,(1 - c^d_{i_1,j_1} + c^a_{i_1,j_1+1}), \end{aligned} \tag{11d}$$

and the meaning is the same as (10) replacing i_1 by i_2 and vice versa.

The second type of collisions during the motion of an agent i_1 is with another agent i_2 that is covering a point. In that case, we calculate the minimum distance between the trajectory of the movement of i_1 and the point where i_2 is covering, $d^{mc}_{i_1.j_1,i_2,j_2}$. If a collision may happen, i.e., $d^{mc}_{i_1.j_1,i_2,j_2} < r_{i_1} + r_{i_2}$, we include the same two sets of constraints as before with the departure and arrival times of the movement of i_1 and the arrival and departure times of i_2 to the point that it must cover. The first set,

$$\begin{aligned} \Delta_{i_1,j_1} - A_{i_2,j_2} &\leq R\,(1 - c^{mc}_{i_1.j_1,i_2,j_2}), \\ \Delta_{i_1,j_1} - \Lambda_{i_1,j_1} &\leq R\,(1 - c^{mc}_{i_1.j_1,i_2,j_2}), \\ \Lambda_{i_1,j_1} - A_{i_2,j_2} &\leq -\varepsilon + R\,(1 - c^{mc}_{i_1.j_1,i_2,j_2}), \\ D_{i_2,j_2} - \Delta_{i_1,j_1} &\leq -\varepsilon + R\,(1 - c^{mc}_{i_1.j_1,i_2,j_2}) \\ &\quad + R\,(1 + c^d_{i_2,j_2} - c^a_{i_2,j_2}), \end{aligned}$$

is active if the movement starts before the coverage and the second,

$$\begin{aligned} A_{i_2,j_2} - \Delta_{i_1,j_1} &\leq R\, c^{mc}_{i_1.j_1,i_2,j_2}, \\ A_{i_2,j_2} - D_{i_2,j_2} &\leq R\, c^{mc}_{i_1.j_1,i_2,j_2}, \\ D_{i_2,j_2} - \Delta_{i_1,j_1} &\leq -\varepsilon + R\, c^{mc}_{i_1.j_1,i_2,j_2}, \\ \Lambda_{i_1,j_1} - A_{i_2,j_2} &\leq -\varepsilon + R\, c^{mc}_{i_1.j_1,i_2,j_2} \\ &\quad + R\,(1 - c^d_{i_1,j_1} + c^a_{i_1,j_1+1}), \end{aligned}$$

if the opposite happens. The interpretation of these constraint sets is the same as for (7)-(8) and (10)-(11).

D. Cost Function

The proposed restrictions guarantee that in the resulting schedule no collisions between agents occur at any time. In some applications, finding such solution may be enough and the problem can be posed as a Constraint Satisfaction Problem. Nevertheless, in many other applications, it is desirable to find a solution that is not only feasible but also optimizes some kind of metric. In particular, we aim to minimize the time in which two or more agents are moving simultaneously:

$$f_{schedule} = \sum_{i_1=1}^{I-1} \sum_{i_2=i_1+1}^{I} \max\big(0, \min(\Lambda_{i_{k_1},j_{k_1}} - \Delta_{i_{k_2},j_{k_2}})\big), \tag{12}$$

with $k_1, k_2 = 1, 2$. $f_{schedule}$ represents the sum of the times in which each pair of movements of different agents are overlapped. This function is motivated by the mobile induction application, that we introduce in Section VI, where the cost function intends to minimize the changes in power requested from the electric grid. Although the cost could also be expressed as a restriction if only one agent were allowed to move at the same time, it is more appropriate to define it as we propose to yield a feasible solution that minimizes simultaneous motion.

The problem of minimizing (12) can be transformed to the standard MILP formulation as follows. First, we introduce the auxiliary variables x_{i_1,j_1,i_2,j_2} to be greater or equal to the minimum inside (12), that is,

$$f_{schedule} = \sum_{i_1=1}^{I-1} \sum_{i_2=i_1+1}^{I} \max(0, x_{i_1,j_1,i_2,j_2}), \tag{13}$$

with

$$\begin{aligned} x_{i_1,j_1,i_2,j_2} \geq \min(&\Lambda_{i_1,j_1} - \Delta_{i_1,j_1}, \Lambda_{i_1,j_1} - \Delta_{i_2,j_2}, \\ &\Lambda_{i_2,j_2} - \Delta_{i_1,j_1}, \Lambda_{i_2,j_2} - \Delta_{i_2,j_2}). \end{aligned}$$

This can be reduced to one of the following constraints

$$x_{i_1,j_1,i_2,j_2} \geq \Lambda_{i_1,j_1} - \Delta_{i_1,j_1} - R\, e^1_{i_1,j_1,i_2,j_2}, \tag{14a}$$
$$x_{i_1,j_1,i_2,j_2} \geq \Lambda_{i_1,j_1} - \Delta_{i_2,j_2} - R\, e^2_{i_1,j_1,i_2,j_2}, \tag{14b}$$
$$x_{i_1,j_1,i_2,j_2} \geq \Lambda_{i_2,j_2} - \Delta_{i_1,j_1} - R\, e^3_{i_1,j_1,i_2,j_2}, \tag{14c}$$
$$x_{i_1,j_1,i_2,j_2} \geq \Lambda_{i_2,j_2} - \Delta_{i_2,j_2} - R\, e^4_{i_1,j_1,i_2,j_2}, \tag{14d}$$
$$e^1_{i_1,j_1,i_2,j_2} + e^2_{i_1,j_1,i_2,j_2} + e^3_{i_1,j_1,i_2,j_2} + e^4_{i_1,j_1,i_2,j_2} = 3, \tag{14e}$$

where $e^k_{i_1,j_1,i_2,j_2}, k \in \{1, \ldots, 4\}$, are binary variables to activate the constraint between (14a) and (14d) that gives the minimum value of x_{i_1,j_1,i_2,j_2}, thanks to Eq. (14e). Note that (14a)-(14d) depend on $c^a_{i,j}$ and $c^d_{i,j}$ through (9) and the difference $\Lambda_{i_{k_1},j_{k_1}} - \Delta_{i_{k_2},j_{k_2}}$ may be different depending on their values, i.e., if any of the movements is split between the end and the beginning of the team plan. For the sake of readability we develop a more detailed description of these constraints in Appendix A.

Second, we introduce the auxiliary variables z_{i_1,j_1,i_2,j_2} to be greater or equal to the maximum inside (12):

$$f_{schedule} = \sum_{i_1=1}^{I-1} \sum_{i_2=i_1+1}^{I} z_{i_1,j_1,i_2,j_2}, \tag{15}$$

such that

$$z_{i_1,j_1,i_2,j_2} \geq 0, \tag{16a}$$

$$z_{i_1,j_1,i_2,j_2} \geq x_{i_1,j_1,i_2,j_2}. \tag{16b}$$

Since the objective (15) is to minimize the sum of z_{i_1,j_1,i_2,j_2}, it can be seen that x_{i_1,j_1,i_2,j_2} will also be minimized by activating only the proper constraint in (14a)-(14d).

E. Optimal Schedule

Finally, we are in position to formulate the complete problem of finding the optimal schedule for the team of agents.

Problem IV.1. *The optimal periodic schedule, in which each agent executes its own periodic plan and which guarantees that no collisions occur and that the time in which two or more movements overlap is minimum, is the solution of the following MILP:*

$$\min f_{schedule} \tag{17}$$

subject to the restrictions introduced between (6) *and* (16).

The variables that define the team schedule are φ_i, which represent the time that each individual plan has to be shifted to produce the team plan. Recall that only $I-1$ of these variables are needed. It is important to emphasize that, although the number of additional variables and restrictions is high, the problem can be solved efficiently using standard state-of-the-art solvers [35]. We provide more details on the computational time in the simulations, Section V. In addition, the solution to the problem is independent of the value of the period since all the times are obtained as a fraction of such period. In fact, the period can be calculated separately depending on the desired performance of the system. For instance, it can be set to the maximum time that a point can remain uncovered or calculated depending on the maximum time allocated to move.

On the downside, there are several reasons for which a feasible solution may not exist. For instance, it may happen if two agents have a common path but in opposite directions, if the space in which the agents can move is limited, or if an agent has to go through a point of the environment that is never left unoccupied by another agent. They can be avoided by modifying the paths of the agents in at least three ways: (i) invert the direction of the movement, (ii) change the points that are assigned to each agent, or (iii) use a different cost function for the path optimization (4). The best solution in each case depends on the application.

V. Simulation Results

In this section we evaluate in simulation the results and computational times of our solution for different cost functions used in the calculation of the optimal times and productions. This function has an important influence in the final paths of the agents and, therefore, in the team plan and the quality of the coverage provided. In particular, we focus on the case of linear functions because they simplify the problem and are appropriate for the application of mobile induction heating, that we introduce in Section VI. The first function that we evaluate is

$$f_1(\theta_{\mathbf{q},r}) = \sum_{i\in\mathcal{I}} \sum_{j\in\Gamma_i} \left(-\theta_{i,j} + \frac{1}{\rho_i^{\max}} \rho_{i,j} \right),$$

in which the goal is to maximize the assigned times and minimize the normalized productions to provide a coverage as homogeneous in time as possible. The second alternative only tries to minimize the productions in order to reduce the maximum actions of the agents:

$$f_2(\theta_{\mathbf{q},r}) = \sum_{i\in\mathcal{I}} \sum_{j\in\Gamma_i} \frac{1}{\rho_i^{\max}} \rho_{i,j},$$

In the third function we include a weight to the times with respect to f_1 that is the inverse of the distance from the agent to the point, normalized by the maximum distance that the agent can reach,

$$f_3(\theta_{\mathbf{q},r}) = \sum_{i\in\mathcal{I}} \sum_{j\in\Gamma_i} \left(-\frac{1}{d_{\mathbf{p}_i^0,\mathbf{q}}} \theta_{i,j} + \frac{1}{\rho_i^{\max}} \rho_{i,j} \right),$$

and in the fourth one we also weight the productions with the relative importance of the required coverage $P^*(\mathbf{q})$ with respect to the maximum production of the agent $\rho_i^{\max}$.

$$f_4(\theta_{\mathbf{q},r}) = \sum_{i\in\mathcal{I}} \sum_{j\in\Gamma_i} \left(-\frac{1}{d_{\mathbf{p}_i^0,\mathbf{q}}} \theta_{i,j} + \frac{P^*(\mathbf{q})}{\rho_i^{\max 2}} \rho_{i,j} \right),$$

In the last two alternatives the times are weighted using the clustering procedure introduced in [26], without the productions in

$$f_5(\theta_{\mathbf{q},r}) = \sum_{i\in\mathcal{I}} \sum_{j\in\Gamma_i} -\omega_{i,j}\, \theta_{i,j},$$

and with weighted productions in

$$f_6(\theta_{\mathbf{q},r}) = \sum_{i\in\mathcal{I}} \sum_{j\in\Gamma_i} \left(-\omega_{i,j}\, \theta_{i,j} + \frac{P^*(\mathbf{q})}{\rho_i^{\max 2}} \rho_{i,j} \right).$$

The evaluation consisted in 10 runs of the algorithm with different positions of the points for different numbers of agents and points, all of them for the six cost functions. The required coverage was randomly selected between 400 and 2200 units while the maximum production of the agents was 5000 units. The number of agents and points was selected in a way that the team had enough production to cover the points. This restricts to at most twice more points than agents.

Since the cost function influences the points that each agent finally visits, we first evaluate in how many cases the optimal solution was found, both for the calculation of times and productions and for the team plan scheduling. In Fig. 2 it can be seen that a solution was found in all the 190 runs for the times and production. However, the first two cost functions only allowed a solution for the scheduling in the 75% and 68% of the cases while the others allowed more than 95% of solutions. This supports the discussion on the existence of solution from in Section IV-E.

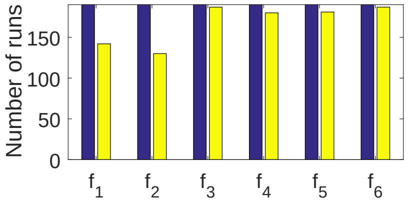


Fig. 2. Number of solutions found for the times and productions (dark blue) and for the scheduling (yellow).

In Fig. 3 we show the computation times to obtain the coverage times and productions solving problem (4). Although in all the cases the increase of the time is more than linear with the number of agents and points, for a team of 13 or 15 agents, the computation requires around 20 seconds in the worst case, i.e., with f_3. This time is negligible for most persistent coverage applications in which the agents are required to carry out the task indefinitely. In addition, the difference between cost functions is significant. The worst-case computation required with f_3 is around 5 times lower with f_5 and in some cases, such as with f_1, the computation increases only with the number of robots.

The results on the computation time of the scheduling are very similar for all the cost function and, therefore, we only show in Fig. 4 the times for f_1. These times are in the order of milliseconds for small teams and only increase to approximately one second for the biggest teams and numbers of points. According to these results, the computational complexity resides in problem (4) rather than in Problem IV.1. Moreover, the small computation time with respect to potential period values supports the recalculation of the solution when $P^*(\mathbf{q})$ changes.

Finally we compare in Table I six different metrics obtained from the simulations to give an idea of the behavior with each function. The first metric is the average number of iterations of Algorithm 1 in [26] to find the final solution. With f_3 and f_5 it is slightly bigger, which means that these cost functions allow the paths to be progressively shortened more than the others. The second metric is the average number of movements per agent during each period, which is really similar for all the functions. The third row includes the average time that each agent dedicates to cover the point in a period. The differences are again small with f_5 and f_6 producing only 2%-better results. In the fourth place we calculate the maximum time that a point remains uncovered every period. The same functions f_5 and f_6 give better results even though there is no much difference between all of them. The fifth metric shows the maximum production required to the agents and normalized by the maximum production available. In this case, f_3 performs much worse than the others while f_4 and f_6 require the smallest maximum productions. Finally we compute a metric that determines the homogeneity of the coverage. It is defined as

$$h = \sum_{\mathbf{q}\in\mathcal{Q}} \sum_{i\in\mathcal{I}} \left(\theta_{i,j} - \frac{P^*(\mathbf{q})}{\rho_i^{\max}} \right)^2 ,$$

and the results show that, on average, the cost functions f_1 and f_2 provide the most homogeneous coverage. As a conclusion, one can see that all the alternatives have advantages and disadvantages and the selection must be made depending on the application.

TABLE I
Comparison of metrics between cost functions.

	f_1	f_2	f_3	f_4	f_5	f_6
Iterations	1.11	**1.10**	1.27	1.12	1.36	1.13
Movements	**1.43**	1.44	1.44	1.44	1.45	1.44
Total Coverage	0.95	0.95	0.94	0.95	**0.97**	**0.97**
Uncovered Time	0.30	0.30	0.32	0.31	**0.29**	**0.29**
Max Production	0.73	0.73	0.93	0.69	0.81	**0.68**
Homogeneity	3.03	**2.93**	3.18	3.16	3.45	3.24

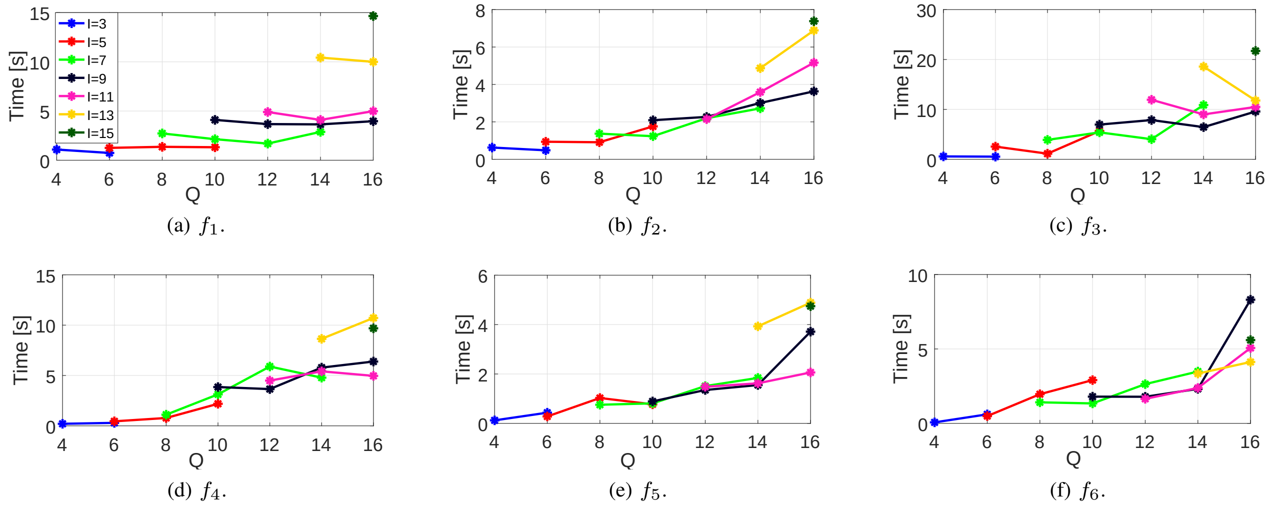
VI. Application to Mobile Induction Heating

Throughout the paper we have presented a solution to persistent coverage with a team of generic agents. Now we particularize the proposal to the application of induction heating in a domestic hob for cooking. In this application, a finite set of cooking pots have to be persistently heated to maintain their temperature or received power as close as possible to the reference determined by the user. This is achieved by means of induction technology with a group of inductors that are capable of transferring power to the pots and can move inside the hob. In particular, we apply our solution to the prototype shown in Fig. 5.

The size of the prototype is 726×482 mm and it has three inductors of 180 mm of diameter. Each one is attached to a robotic arm with two rotational joints. This design allows two configurations of the arms in many positions of the inductors, one with the second joint to the left of the inductor and another with the joint to the right. However, when moving closer to the hob limits, only one configuration may be feasible without colliding with the border and to change between configurations the inductor must go through its home position.

The motion of the arms is given by stepper motors, which are controlled with Arduino boards and specific drivers. The strategies are computed in Matlab and transmitted to the Arduino boards through the serial bus. Since the interest of this validation is in the movements of the inductors and the periodic strategy, in these experiments the pots have been simulated and the power has not been actually provided in order to accelerate the experimentation and guarantee the safety of the motion system.

In this context, the discrete points represent the pots, the coverage level is their received power and the agents are the inductors. The coverage level deteriorates over time because the pots cool down if they are not heated. The maximum



(a) f_1.

(b) f_2.

(c) f_3.

(d) f_4.

(e) f_5.

(f) f_6.

Fig. 3. Mean calculation time of the coverage times and productions solving problem (4). Each figure represents the results with a different cost function and different colors represent different number of agents.

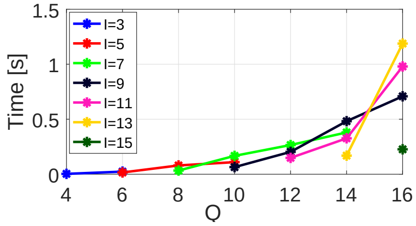


Fig. 4. Mean calculation time of the scheduling solving Problem IV.1.

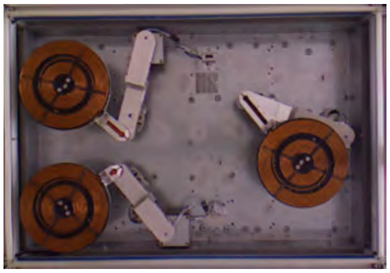


Fig. 5. Prototype of induction hob with mobile inductors.

production depends on the design of the inductor and the coupling pot-inductor and the inductors may not be able to reach all the pots due to the physically-limited reachability of the robotic arms.

The cost function that we have proposed for the team plan scheduling, that aims to minimize the time in which two or more agents are moving simultaneously, is motivated by this application as follows. When an inductor starts moving from one pot to the following, it stops heating and, therefore, it stops requesting power from the electric grid. In the same way, when it reaches a new pot and starts heating, it starts requesting power from the grid. These changes in requested power are limited by the European norm UNE-EN 61000-3-3, 2013. In order to comply with this norm, we minimize the time of simultaneous movements so that, when an inductor stops (starts) requesting power, we can increase (decrease) the request of the others to compensate the total request of the system.

Given the particular features of the prototype and its spatial restrictions, there may appear cases in which the scheduling may not have a feasible solution due to the collision avoidance restrictions. For instance, if an inductor is in charge of only one pot, i.e., it remains all the time under such pot, and another inductor has to heat a different pot sufficiently close to the first one, then the second inductor may not be capable of reaching it without colliding with the first one. To solve these issues, geometric solutions have been implemented ad hoc and, since we focus on the general approach to periodic persistent coverage, it is out of the scope of this paper.

For this particular application we calculate the period as $T = \max_{i \in \mathcal{I}} t_i^m / \theta_{\max}^m$, where t_i^m are the actual times that the robots need to traverse the path and $\theta_{\max}^m$ is the maximum portion of the period that we allow for the motion, in this case, 0.3.

A. *Experimental Results*

We provide experimental results of the periodic heating strategy with the prototype introduced before. The results obtained for different configurations of the pots are conceptually the same and, therefore, we only show an example.

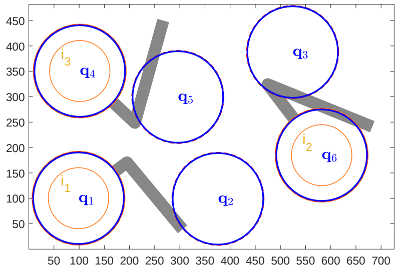


Fig. 6. Layout of one of the experiments. Blue circumferences represent the pots; orange circumferences, the two coils of the inductors; and grey segments, the two links of the robotic arms.

In Fig. 6 the layout of the example is shown. There are 6 virtual pots which are circles of diameter $18cm$. The pots require a constant $P^*(\mathbf{q}) = 800, 1500, 500, 1500, 2000$ and $1500W$, respectively. At this particular time instant, the three inductors are heating their assigned pots.

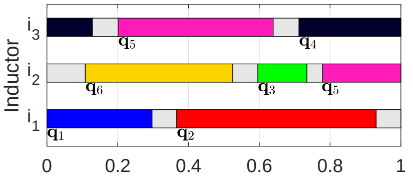


Fig. 7. Optimal schedule obtained for the experiment.

The optimal schedule obtained for this configuration of inductors and pots is shown in Fig. 7. It can be seen that the inductor i_2, the right one in Fig. 5, has to spend most of its time heating pots $\mathbf{q}_3$ and $\mathbf{q}_6$ and also move to heat pot $\mathbf{q}_5$ in cooperation with inductor i_3, the top one, that is also in charge of pot $\mathbf{q}_4$. On the other hand, inductor i_1 alternatively heats pots $\mathbf{q}_1$ and $\mathbf{q}_2$.

The power received by the pots can be seen in Fig. 8a-8f. One can see that all the pots receive power periodically for a period of time and then have to wait some time until the following coverage. This is a requirement of the system since there are more pots than inductors. However, the average received power quickly tends to the required power, meaning that, on average, the pots receive the desired power.

It is important to note that, although the curly power profile may result in important variations of the temperature of the pot, this effect can only be negative in boiling processes, where the bubbles can appear and disappear with the power changes. In other processes, such as deep frying or just warming, the variation of the temperature can hardly be appreciated by the user. Moreover, if a boiling process is taking place, the assignment of the inductors can be adjusted to keep an inductor under the boiling pot all the time while the other share the rest of the pots.

In Fig. 9 the power provided by the inductors is depicted along with the total power of the hob. The cost function of the scheduling, that minimizes the time in which more than one inductor is moving at the same time, favors that the variations of the total power are minimal. In addition, it allows us to adjust the total power of the hob when one inductor starts or stops requesting power with the other two inductors to satisfy the flicker norm.

Finally, we carried out an evaluation of all the possible solutions for this experiment using brute force. In particular, we calculated all the possible combinations of trajectories of the three inductors and tried to solve the second and fourth steps of Algorithm 1. Out of the 1270 combinations, only 10 were feasible and the solution provided by our method was only a 1.41% away from the best one in terms of coverage quality measured by $\int_0^t \sum_{\mathbf{q}\in\mathcal{Q}} \left(Z^*(\mathbf{q},\tau) - Z(\mathbf{q},\tau)\right)^2 d\tau,$. Finding such best solution took 181s while our method only needed 0.95s.

VII. Conclusions

In this paper we have introduced a periodic solution to the persistent coverage problem. We have proposed a path planning strategy for each individual agent and a quadratically constrained linear program to obtain the optimal coverage times and actions to satisfy the coverage objective of the environment. We have also presented an MILP to find a periodic team schedule in which no collisions occur. Simulation results support our proposal and demonstrate that it is tractable for significant team and environment sizes. Moreover, we have shown experimental results for a heating application in which a set of pots is homogeneously heated in a domestic hob with mobile inductors.

Appendix A
Cost Function Constraints

The constraints that arise from the minimum function in (12) capture the duration of the overlap between each pair of movements. In the case that one of the movements is completely overlapped with the other, the duration of the overlap is equal to the duration of such movement. Therefore, we can express Eq. (14a) and (14d) as

$$x_{1,2} \geq \Lambda_1 - \Delta_1 - R\, e^1_{1,2},$$
$$x_{1,2} \geq \Lambda_2 - \Delta_2 - R\, e^4_{1,2}.$$

In this appendix, we substitute subindices i_1, j_1 and i_2, j_2 by 1 and 2 for simplicity.

On the other hand, if the movements are only partially overlapped, the duration of such overlap is bounded by Eq. (14b) or Eq. (14c). However, the differences $\Lambda_1 - \Delta_2$ or $\Lambda_2 - \Delta_1$ may not represent such duration if one or both movements are split between the end and the beginning of the team plan, that happens if $c^d_k \equiv c^d_{i_k,j_k} = 0$ and $c^a_i \equiv c^a_{i_k,j_k+1} = 1$. In the following we formulate the restrictions for all the cases in which none, one or both movements are split. Each constraint is conditionally activated, applying the big number method, depending on c^d_k and c^a_k. In particular, if $c^a_k - c^d_k = 0$, the movement is not split, and its corresponding constraints activated, and if $1 - c^a_k + c^d_k = 0$, the movement is split.

We set these two restrictions for the case in which none of the movements is split:

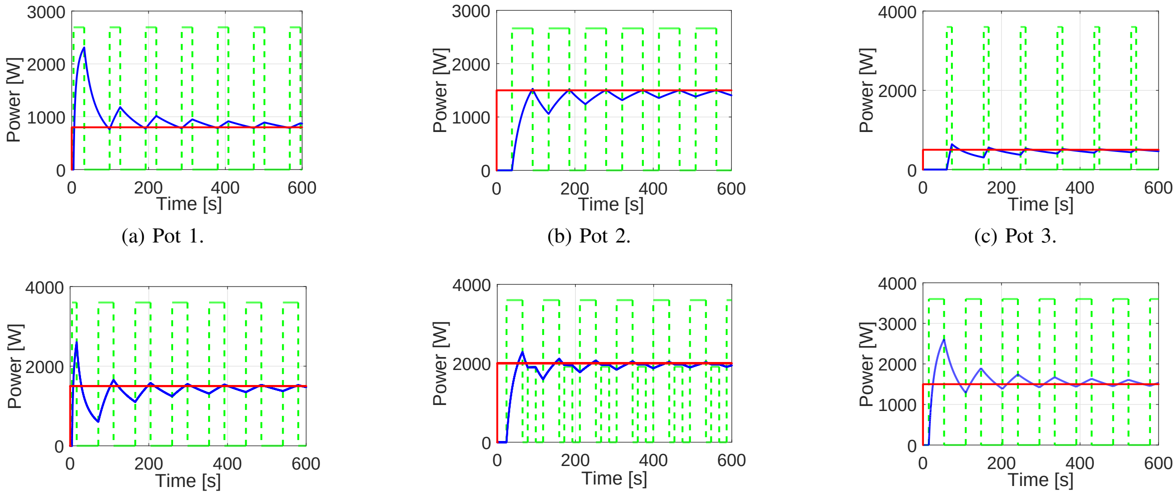
$$x_{1,2} \geq \Lambda_1 - \Delta_2 - R\, e^1_{1,2} - R\,(c^a_1 - c^d_1) - R\,(c^a_2 - c^d_2),$$
$$x_{1,2} \geq \Lambda_2 - \Delta_1 - R\, e^4_{1,2} - R\,(c^a_1 - c^d_1) - R\,(c^a_2 - c^d_2)$$

Similarly, when both are split, the constraints are

$$\begin{aligned} x_{1,2} \geq \Lambda_1 - (\Delta_2 - 1) - R\, e^1_{1,2} &- R(1 - c^a_1 - c^d_1) \\ &- R\,(1 - c^a_2 - c^d_2), \\ x_{1,2} \geq \Lambda_2 - (\Delta_1 - 1) - R\, e^4_{1,2} &- R(1 - c^a_1 - c^d_1) \\ &- R\,(1 - c^a_2 - c^d_2). \end{aligned}$$

In this case we only subtract 1 to the initial times to overcome the split.

We proceed in a similar way when only one of the movements is split although in these cases it is slightly more complicated. In the first place we focus in the case in which



(a) Pot 1.

(b) Pot 2.

(c) Pot 3.

(d) Pot 4.

(e) Pot 5.

(f) Pot 6.

Fig. 8. Power received by the pots during the experiment. The green line represents the instant received power; the red line, the requested power; and the blue line, the average received power.

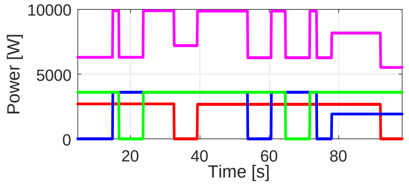


Fig. 9. Power provided by the inductors during the experiment. The red, blue and green lines represent the power of each inductor and the magenta line, the total power.

the movement of agent 1 is split. We check which part of the movement may overlap with the movement of agent i_2. This is done by introducing a binary variable $b^1_{1,2}$ such that $b^1_{1,2} = 1$ if $\Delta_1 \leq \Lambda_2$, meaning that the second part is overlapped, and $b^1_{1,2} = 0$, otherwise, meaning that the candidate for overlap is the first part. This variable is obtained from the following constraints:

$$\begin{aligned}\Delta_1 - \Lambda_2 &\leq R\,(1 - c^a_1 - c^d_1) + R\,(c^a_2 - c^d_2) + R\,(1 - b^1_{1,2}),\\ \Lambda_2 - \Delta_1 &\leq R\,(1 - c^a_1 - c^d_1) + R\,(c^a_2 - c^d_2) + R\,b^1_{1,2}.\end{aligned}$$

Depending on the value of $b^1_{1,2}$ we transform the constraints differently. When $b^1_{1,2} = 1$ we add one to the final time of the movement of i_1. This can be seen as moving the part of the movement that is at the beginning of the plan to the right side. Thus, the constraints are

$$\begin{aligned}x_{1,2} \geq{}& (\Lambda_1 + 1) - \Delta_2 - R\,e^1_{1,2} - R\,(1 - c^a_1 - c^d_1)\\ &- R\,(c^a_2 - c^d_2) - R\,(1 - b^1_{1,2}),\\ x_{1,2} \geq{}& \Lambda_2 - \Delta_1 - R\,e^4_{1,2} - R\,(1 - c^a_1 - c^d_1)\\ &- R\,(c^a_2 - c^d_2) - R\,(1 - b^1_{1,2}).\end{aligned}$$

On the contrary, when $b^1_{1,2} = 0$, we subtract one to the initial time of the movement to compare the movement of agent i_2 with the left part of the movement of i_1. The constraints result in

$$\begin{aligned}x_{1,2} \geq{}& \Lambda_1 - \Delta_2 - R\,e^1_{1,2} - R\,(1 - c^a_1 - c^d_1)\\ &- R\,(c^a_2 - c^d_2) - R\,b^1_{1,2},\\ x_{1,2} \geq{}& \Lambda_2 - (\Delta_1 - 1) - R\,e^4_{1,2} - R\,(1 - c^a_1 - c^d_1)\\ &- R\,(c^a_2 - c^d_2) - R\,b^1_{1,2}.\end{aligned}$$

The same procedure can be followed for the case in which the movement of i_2 is split. We include the binary variable $b^2_{1,2}$ with

$$\begin{aligned}\Delta_2 - \Lambda_1 &\leq R\,(c^a_1 - c^d_1) + R\,(1 - c^a_2 - c^d_2) + R\,(1 - b^2_{1,2}),\\ \Lambda_1 - \Delta_2 &\leq R\,(c^a_1 - c^d_1) + R\,(1 - c^a_2 - c^d_2) + R\,b^2_{1,2}.\end{aligned}$$

When $b^2_{1,2} = 1$, the constraints are

$$\begin{aligned}x_{1,2} \geq{}& \Lambda_1 - \Delta_2 - R\,e^1_{1,2} - R\,(c^a_1 - c^d_1)\\ &- R\,(1 - c^a_2 - c^d_2) - R\,(1 - b^2_{1,2}),\\ x_{1,2} \geq{}& (\Lambda_2 + 1) - \Delta_1 - R\,e^4_{1,2} - R\,(c^a_1 - c^d_1)\\ &- R\,(1 - c^a_2 - c^d_2) - R\,(1 - b^2_{1,2}),\end{aligned}$$

and, otherwise,

$$\begin{aligned}x_{1,2} \geq{}& \Lambda_1 - (\Delta_2 - 1) - R\,e^1_{1,2} - R\,(c^a_1 - c^d_1)\\ &- R\,(1 - c^a_2 - c^d_2) - R\,b^2_{1,2},\\ x_{1,2} \geq{}& \Lambda_2 - \Delta_1 - R\,e^4_{1,2} - R\,(c^a_1 - c^d_1)\\ &- R\,(1 - c^a_2 - c^d_2) - R\,b^2_{1,2}.\end{aligned}$$

Finally, note that for the implementation and resolution of the problem, the constraints (14a)-(14d) have to be substituted by all the restrictions of this appendix.

References

[1] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," *Studies in Computational Intelligence*, vol. 604, pp. 31–51, 2015.

[2] I. I. Hussein and D. M. Stipanovic, “Effective Coverage Control for Mobile Sensor Networks With Guaranteed Collision Avoidance,” *IEEE Trans. on Control Systems Technology*, vol. 15, no. 4, pp. 642–657, 2007.

[3] N. M. Kakalis and Y. Ventikos, “Robotic swarm concept for efficient oil spill confrontation,” *J. Hazardous Materials*, vol. 154, no. 1, pp. 880–887, 2008.

[4] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, “Persistent ocean monitoring with underwater gliders: Adapting sampling resolution,” *J. Field Robotics*, vol. 28, no. 5, pp. 714–741, 2011.

[5] M. Jadaliha and J. Choi, “Environmental monitoring using autonomous aquatic robots: Sampling algorithms and experiments,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 899–905, 2013.

[6] D. Mitchell, N. Chakraborty, K. Sycara, and N. Michael, “Multi-Robot Persistent Coverage with stochastic task costs,” in *IEEE/RSJ Int. Conf. Intell. Robots Systems*. IEEE, 2015, pp. 3401–3406.

[7] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, 2004.

[8] P. F. Hokayem, D. Stipanovic, and M. W. Spong, “On persistent coverage control,” in *IEEE Conf. Decision and Control*, 2007, pp. 6130–6135.

[9] J. M. Palacios-Gasós, E. Montijano, C. Sagüés, and S. Llorente, “Distributed coverage estimation and control for multi-robot persistent tasks,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1444–1460, 2016.

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

[11] J. M. Palacios-Gasos, E. Montijano, C. Sagüés, and S. Llorente, “Multi-robot persistent coverage using branch and bound,” in *American Control Conf.*, 2016, pp. 5697–5702.

[12] X. Lin and C. G. Cassandras, “An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces,” in *IEEE Conf. Decision and Control*, 2013, pp. 6886–6891.

[13] X. Lan and M. Schwager, “Planning periodic persistent monitoring trajectories for sensing robots in gaussian random fields,” in *IEEE Int. Conf. Robot. Autom.*, 2013, pp. 2407–2412.

[14] ——, “Rapidly Exploring Random Cycles: Persistent Estimation of Spatiotemporal Fields With Multiple Sensing Robots,” *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1230–1244, 2016.

[15] D. Portugal, C. Pippin, R. P. Rocha, and H. Christensen, “Finding optimal routes for multi-robot patrolling in generic graphs,” in *IEEE/RSJ Int. Conf. Intell. Robots and Systems*, 2014, pp. 363–369.

[16] C. Song, L. Liu, G. Feng, Y. Wang, and Q. Gao, “Persistent awareness coverage control for mobile sensor networks,” *Automatica*, vol. 49, no. 6, pp. 1867–1873, 2013.

[17] G. A. Korsah, A. Stentz, and M. B. Dias, “A comprehensive taxonomy for multi-robot task allocation,” *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[18] K. Schüpbach and R. Zenklusen, “An adaptive routing approach for personal rapid transit,” *Mathematical Methods of Operations Research*, vol. 77, no. 3, pp. 371–380, 2013.

[19] J. Kim, B. D. Song, and J. R. Morrison, “On the scheduling of systems of UAVs and fuel service stations for long-term mission fulfillment,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 70, no. 1-4, pp. 347–359, 2013.

[20] K. Sundar and S. Rathinam, “Algorithms for Routing an Unmanned Aerial Vehicle in the Presence of Refueling Depots,” *IEEE Trans. on Automation Science and Engineering*, vol. 11, no. 1, pp. 287–294, 2014.

[21] J. L. Fargeas, B. Hyun, P. Kabamba, and A. Girard, “Persistent visitation under revisit constraints,” in *International Conference on Unmanned Aircraft Systems*, 2013, pp. 952–957.

[22] D. Mitchell, M. Corah, N. Chakraborty, K. Sycara, and N. Michael, “Multi-robot long-term persistent coverage with fuel constrained robots,” in *IEEE Int. Conf. Robot. Autom.*, 2015, pp. 1093–1099.

[23] J. Yu, M. Schwager, and D. Rus, “Correlated Orienteering Problem and its Application to Persistent Monitoring Tasks,” *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1106–1118, 2016.

[24] N. Nigam, “The Multiple Unmanned Air Vehicle Persistent Surveillance Problem: A Review,” *Machines*, vol. 2, no. 1, pp. 13–72, 2014.

[25] S. L. Smith, M. Schwager, and D. Rus, “Persistent robotic tasks: Monitoring and sweeping in changing environments,” *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 410–426, 2012.

[26] J. M. Palacios-Gasós, E. Montijano, C. Sagüés, and S. Llorente, “Multi-robot persistent coverage with optimal times,” in *Conf. Decision and Control*, 2016, pp. 3511–3517.

[27] J. Yu, J. Aslam, S. Karaman, and D. Rus, “Anytime planning of optimal schedules for a mobile sensing robot,” in *IEEE/RSJ Int. Conf. Intell. Robots Systems*, 2015, pp. 5279–5286.

[28] X. Wang, M. Kloetzer, C. Mahulea, and M. Silva, “Collision avoidance of mobile robots by using initial time delays,” in *IEEE Conf. Dec. Control*, 2015, pp. 324–329.

[29] M. Turpin, K. Mohta, N. Michael, and V. Kumar, “Goal assignment and trajectory planning for large teams of interchangeable robots,” *Autonomous Robots*, vol. 37, no. 4, pp. 401–415, 2014.

[30] N. Christofides, “Worst-case analysis of a new heuristic for the travelling salesman problem,” Graduate School of Industrial Administration, Carnegie Mellon University, Tech. Rep., 1976.

[31] C. Prins, P. Lacomme, and C. Prodhon, “Order-first split-second methods for vehicle routing problems: A review,” *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 179–200, 2014.

[32] D. Richert and J. Cortes, “Robust distributed linear programming,” *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2567–2582, 2015.

[33] E. Montijano and A. R. Mosteo, “Efficient multi-robot formations using distributed optimization,” in *IEEE Conf. Decision and Control*, 2014, pp. 6167–6172.

[34] I. Griva, S. G. Nash, and A. Sofer, *Linear and nonlinear optimization*. SIAM, 2009.

[35] “IBM-ILOG: CPLEX optimization studio.” [Online]. Available: http://www.ilog.com/products/cplex

José Manuel Palacios-Gasós obtained his Industrial Engineering degree (mention in Industrial Automation and Robotics) in 2012 and his M.Sc. and Ph.D. degree in Systems Engineering and Computer Science from the University of Zaragoza in 2014 and 2018, respectively. His current research interests include multi-agent systems, distributed and cooperative control and computer vision. In 2018 he joined BSH Home Appliances Group, where he is a member of the Firmware Group in Induction Cooktops Technology.

Eduardo Montijano (M’12) is an assistant professor at Universidad de Zaragoza, Spain. He received the M.Sc. and Ph.D. degrees from the Universidad de Zaragoza, Spain, in 2008 and 2012 respectively, obtaining the extraordinary award of the Universidad de Zaragoza in the 2012–2013 academic year. He was also a faculty member at Centro Universitario de la Defensa, Zaragoza, Spain, between 2012 and 2016. His main research interests include distributed algorithms, cooperative control and computer vision.

Carlos Sagüés (M’00-SM’11) received the M.Sc. and Ph.D. degrees from the Universidad de Zaragoza, Spain. During the course of his Ph.D. he worked on force and infrared sensors for robots. Since 1994 he has been Associate Professor and, since 2009, Full Professor with the Departamento de Informatica e Ingeniería de Sistemas, Universidad de Zaragoza, where he has also been Head Teacher. His current research interest includes control systems, computer vision, visual robot navigation and multivehicle cooperative control.

Sergio Llorente received the M.Sc. and Ph.D. degrees in electronics engineering from the University of Zaragoza in 2001 and 2016, respectively. In 2001, he joined BSH Home Appliances Group where he has held different positions in the Research and Development Department of Induction Cooktops. He is currently in charge of several research lines and is an Inventor in more than 150 patents. He has also been an Assistant Professor with the University of Zaragoza since 2004. His research interests include power electronics, simulation and control algorithms for power electronics, and temperature control.