Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Distributed Multirobot Path Planning in Unknown Maps Using Petri Net Models ⋆

Cristian Mahulea *, **Eduardo Montijano** *, **Marius Kloetzer** **

* *Aragón Institute of Engineering Research (I3A), University of Zaragoza, Spain (e-mail: {cmahulea,emonti}@ unizar.es).*
** *"Gheorghe Asachi" Technical University of Iasi, Romania (e-mail: kmarius@ac.tuiasi.ro)*

Abstract: This paper considers the path planning problem in multirobot systems with an unknown environment. The robots' mission is given as a Boolean formula on the final states. We assume that the robots have partial knowledge of the environment and they are able to estimate the environment using a recursive Bayes estimator. Furthermore, they communicate between them if they are at a distance smaller than a given threshold in order to improve their own estimation. Each robot will solve an optimization problem based on the Petri net model of the environment and it will move accordingly. We provide an algorithm to be iterated by each robot and we evaluate the results by simulation.

Keywords: discrete event systems, path planning, environment estimation, multirobot systems

1. INTRODUCTION

Planning the path of mobile robots is an actual problem that continues to receive a lot of attention. Starting from various methods for solving the classical planning of one robot in a known environment cluttered with obstacles (Latombe, 1991; Choset et al., 2005; LaValle, 2006; Mahulea et al., 2020), researchers have proposed different scenarios that prove useful as mobile robots get involved in multiple applications. A few examples of such scenarios include high level specifications that express the desired mission to be accomplished (Belta et al., 2007; Fainekos et al., 2009) or planning teams of multiple robots in centralized or decentralized manners (Mahulea and Kloetzer, 2018; Guo et al., 2014). In many scenarios, the robotic workspace is partitioned in a finite number of regions (or cells) by using existing techniques (Berg et al., 2008; Choset et al., 2005), and the problem can be solved on an implied discrete-event representation for example using methods characteristic to discrete systems or Harmonic functions Garrido et al. (2010).

In case of workspaces that are not fully known, the proposed problems range from static environments with probabilistic information on the existence of regions or obstacles (Ding et al., 2014; Svorenova et al., 2012; Kloetzer and Mahulea, 2015; Kavraki et al., 1996) to simultaneous localization and mapping in dynamic environments (Huang et al., 2005; Zamora and Yu, 2014). An important scenario is the one in which the robots have a limited communication range and whenever possible they exchange their information regarding the environment with the purpose of eventually attaining a collective desired behavior.

In this line of work, this paper combines an approach where a team of robots is planned based on a Boolean formula, Mahulea and Kloetzer (2018), together with a distributed inference algorithm, Julian et al. (2012), and a polynomial-based consensus method, Montijano et al. (2013). The method from (Mahulea and Kloetzer, 2018) assumes an initially fully-known environment where some regions of interest exists and a centralized team of cooperating mobile robots. A Boolean-based formula is imposed for the whole team, expressing a desired behavior on both robot trajectories and on their final positions, and the individual plans for the robots are found by using a Petri net model of the team and mathematical programming techniques that embed this model and the formula. The method from Julian et al. (2012) presents a distributed inference algorithm to compute the probability distribution of the class associated to each region. Since the algorithm relies on a consensus iteration, in order to reduce the number of communication rounds we use the consensus algorithm described in Montijano et al. (2013) to exploit the good convergence properties of Chebyshev polynomials. The combination of these techniques allows the team of robots to satisfy the Boolean formula without requiring complex synchronization and/or communication mechanisms.

In short, this paper assumes a static but *initially unknown* environment in which some regions of interest exist, a team of robots with restricted communication radius, and a Boolean formula on the set of regions of interest. For simplicity of exposition, the formula expresses a requirement on the final (stopping) position of the robots, e.g., in what regions the robots should or should not stop, by ignoring requirements on trajectories. Moreover, the environment is assumed already partitioned, such that the robots are regarded as moving on a graph-like discrete event model.

⋆ The work of C. Mahulea has been partially supported by the MINECO "Salvador de Madariaga" program. This work was partly supported by ONRG-NICOP-grant N62909-19-1-2027. M. Kloetzer acknowledges the CNCS-UEFISCDI grant PN-III-P1-1.1-TE-2016-0737.

Each robot can obtain noisy observations of the state (observation) of the partition cells and uses a consensus method to improve the estimation, exploiting other robots information. These estimations are used as input of an optimization problem that provides the motion commands to the robots. The whole team repeats the estimation and planning steps until satisfaction of the Boolean formula, a condition that is locally verified.

The rest of the paper is structured as follows. Section 2 presents the notation and formulates the problem. Sections 3 and 4 include some aspects related to Boolean-based planning in known environments and to estimation of the environment based on local information, respectively. Section 5 combines these methods, obtaining a new planning and estimation algorithm applicable to initially unknown environments. Section 6 shows simulation results, while section 7 draws some concluding remarks.

2. PROBLEM DEFINITION

Let us assume a partitioned environment such that $P = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$ is the set of disjoint regions (cells) composing the environment, where $|\cdot|$ denotes the cardinality of a set, i.e., the number of elements. Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_{|\Pi|}\}$ be a set of atomic propositions such that $\pi_i \in \Pi$ is the label corresponding to the region of interest i, e.g., color i. Each cell of the environment is labeled with at most one label from the set Π through a function $h : P \to \{\Pi \cup \{\emptyset\}\}$. For all $p_i \in P$, $h(p_i)$ is the label of region p_i and we say that p_i is a free-space and not a region of interest if $h(p_i) = \emptyset$, i.e., if it is labeled by the empty symbol.

In this environment, a number of $|R|$ identical robots evolve, where $R = \{r_1, r_2, \ldots, r_{|R|}\}$ is the set of robots. The robots know the set P of partitions elements and the adjacency of the regions in this set, but they do not know the labels of the regions, i.e., they do not know the function h. We assume that they have a probabilistic sensing measurement of the environment and they are capable of estimating for each region p_i its label. Moreover, if a robot is in a region p_i we assume that it can identify with probability 1 the type of region.

The specification for the robots is given as a Boolean formula defined over the variables from set Π. The specification is global and should be satisfied in the final team position.

Local communications between robots occur when they are at a distance less than a given threshold. They interchange the information of the probabilistic estimation of the function h. After communicating, each robot solves locally a path planning problem and moves to the next region. If the path planning problem is unfeasible, due to the incomplete knowledge of the environment, it chooses randomly an adjacent region and moves. We assume that in each region at most one robot may exist at a given time.

The problem we want to solve in this paper is to ensure the fulfilment of the Boolean formula at the final states, i.e., when all robots stop.

3. BOOLEAN BASED PLANNING

For solving the path planning problem, Petri nets models will be used (Mahulea and Kloetzer, 2018) that provide, in general, solutions for teams with more robots than the standard approaches based on transition system or Markov Decision Process models (Kloetzer and Mahulea, 2020). Assuming the sets in Section 2, the Petri net model is defined as follows.

Definition 1. A Robot Motion Petri Net (RMPN) system is a tuple $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}, \Pi, h, \boldsymbol{m}_0 \rangle$, where

- $P = \{p_1, p_2, \ldots, p_{|P|}\}$ is a finite set of places, each place $p_i \in P$ is modeling the region $p_i \in P$.
- $T = \{t_1, t_2, \ldots, t_{|T|}\}$ is a finite set of transitions. If regions p_i and p_j are adjacent, then there exists two transitions $t_{i,j}$ and $t_{j,i}$ in T modeling the movement from p_i to p_j and from p_j to p_i, respectively.
- $\boldsymbol{Pre} \in \mathbb{N}^{|P| \times |T|}$ is the pre incidence matrix containing the weights of the arcs connecting places to transitions. In particular, $Pre[p_i, t_j] = 1$ if there exists an arc from p_i to t_j of weight 1.
- $\boldsymbol{Post} \in \mathbb{N}^{|P| \times |T|}$ is the post incidence matrix containing the weights of the arcs connecting transitions to places. In particular, $Post[p_i, t_j] = 1$ if there exists an arc from t_j to p_i of weight 1.
- $\Pi \cup \{\emptyset\}$ is the set of output symbols (observations), where $\emptyset$ denotes the empty observation.
- $h : P \to \{\Pi \cup \{\emptyset\}\}$ is the observation function where $h(p_i)$ is the output of place $p_i \in P$.
- $\boldsymbol{m}_0 \in \mathbb{N}^{|P|}$ is the initial marking such that $m_0[p_i]$ is the number of robots initially located in p_i.

For a node $x \in P \cup T$, ${}^\bullet x$ denotes the set of input nodes while $x^\bullet$ denotes the set of output nodes of x. Formally, for $p_i \in P$, ${}^\bullet p_i = \{t_j \in T | Post[p_i, t_j] > 0\}$ and $p_i{}^\bullet = \{t_j \in T | Pre[p_i, t_j] > 0\}$ while for a $t_j \in T$, ${}^\bullet t_j = \{p_i \in P | Pre[p_i, t_j] > 0\}$ and $t_j{}^\bullet = \{p_i \in P | Pre[p_i, t_j] > 0\}$ respectively.

The marking (state) of the RMPN may change by firing the transitions. A transition t_j may fire at a given marking $\boldsymbol{m}_k$ if it is enabled: $\forall p_i \in {}^\bullet t_j$, $m_k[p_i] \geq Pre[p_i, t_j]$. If transition t_j is fired at $\boldsymbol{m}_k$, the new marking that is obtained is given by,

$$\boldsymbol{m}_l = \boldsymbol{m}_k + \boldsymbol{C}[\cdot, t_j],$$

where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the incidence matrix and $\boldsymbol{C}[\cdot, t_j]$ is the column corresponding to t_j. We say that the marking $\boldsymbol{m}_l$ is reachable from marking $\boldsymbol{m}_k$. If a sequence of transitions σ is fired from $\boldsymbol{m}_k$, defining the Parikh vector $\boldsymbol{\sigma} \in \mathbb{N}^{|T| \times 1}$ in which each element count the number of firings of each transition in the sequence σ, then the reachable marking $\boldsymbol{m}$ that is reached after firing σ satisfies,

$$\boldsymbol{m} = \boldsymbol{m}_k + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \tag{1}$$

Equation (1) is called state or fundamental equation of the RMPN. Notice that (1) is only a necessary condition for the reachability of a marking $\boldsymbol{m}$. A solution $\langle \boldsymbol{m}, \boldsymbol{\sigma} \rangle$ satisfying (1) for a given initial marking $\boldsymbol{m}_k$ is not necessarily a reachable marking $\boldsymbol{m}$, neither $\boldsymbol{\sigma}$ corresponds to a sequence of transitions that can be fired.

For a given observation $\pi_i \in \Pi$, let us define its characteristic vector $\boldsymbol{v}_i \in \mathbb{N}^{1 \times |P|}$ such that, $v_i[p_k] = 1$ if $h(p_k) = \pi_i$

and $v_i[p_k] = 0$ otherwise. Notice that for a given reachable marking $\boldsymbol{m}$, if $\boldsymbol{v}_i \cdot \boldsymbol{m} \geq 1$, then observation π_i is active at $\boldsymbol{m}$, i.e., there is at least one robot in a region labeled as π_i at $\boldsymbol{m}$. From a given initial marking $\boldsymbol{m}_0$, if we would like to compute a final marking $\boldsymbol{m}$ in which observation π_i is active, the following constraints should be satisfied:

$$\begin{cases} \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma} \\ \boldsymbol{v}_i \cdot \boldsymbol{m} \geq 1 \end{cases}$$

Let us denote by φ the Boolean formula that the robots should fulfill at the final states and let us define for each atomic proposition $\pi_i \in \Pi$ a Boolean variable x_i such that $x_i = 1$ is π_i is active, otherwise $x_i = 0$. Furthermore, let $\boldsymbol{x} \in \{0,1\}^{|\Pi|\times 1}$ be a vector containing all variables x_i. It is possible to define a set of linear inequalities using variables $\boldsymbol{x}$ such that if a vector $\boldsymbol{x}$ is a solution, i.e., it satisfies the inequalities, then the corresponding active regions according to $\boldsymbol{x}$ satisfy φ. In (Mahulea and Kloetzer, 2018), an algorithm is presented such that for a given Boolean formula φ, the set of linear inequalities is computed. Let us denote these inequalities as,

$$\boldsymbol{A}_{task} \cdot \boldsymbol{x} \leq \boldsymbol{b}_{task}. \tag{2}$$

Furthermore, for a given solution $\boldsymbol{x}$ of (2), if $x_i = 1$, then observation π_i should be active at the final marking. Assuming that $\boldsymbol{m}$ is the final marking, the following rule should be satisfied:

If $x_i = 1$ *then* $\boldsymbol{v}_i \cdot \boldsymbol{m} \geq 1$ *else* $\boldsymbol{v}_i \cdot \boldsymbol{m} = 0$

This can be tackled by introducing the following two constraints (in which N is a big number):

$$\begin{cases} N \cdot x_i \geq \boldsymbol{v}_i \cdot \boldsymbol{m} \\ \quad x_i \leq \boldsymbol{v}_i \cdot \boldsymbol{m} \end{cases} \tag{3}$$

Notice that, if $x_i = 1$, the first inequality is always satisfied ($N \geq \boldsymbol{v}_i \cdot \boldsymbol{m}$ being N a big number) while the second inequality forces $\boldsymbol{v}_i \cdot \boldsymbol{m}$ to be greater than or equal to 1 implying that π_i is active at $\boldsymbol{m}$. On the contrary, if $x_i = 0$, the second constraint is always satisfied (since $\boldsymbol{v}_i$ and $\boldsymbol{m}$ are positive vectors) while the first one becomes $0 \geq \boldsymbol{v}_i \cdot \boldsymbol{m}$ that is satisfied only if $\boldsymbol{v}_i \cdot \boldsymbol{m} = 0$ and π_i will not be active at $\boldsymbol{m}$.

Putting (1), (2) and (3) together, in order to compute a final state $\boldsymbol{m}$ where the formula φ is satisfied, the following Mixed Integer Linear Programming (MILP) optimization problem can be solved:

$$\begin{aligned} &\min \mathbf{1}^T \cdot \boldsymbol{\sigma} \\ &s.t. \begin{cases} \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma} \\ \boldsymbol{A}_{task} \cdot \boldsymbol{x} \leq \boldsymbol{b}_{task} \\ N \cdot \boldsymbol{x} \geq \boldsymbol{V} \cdot \boldsymbol{m} \\ \boldsymbol{x} \leq \boldsymbol{V} \cdot \boldsymbol{m} \end{cases} \end{aligned} \tag{4}$$

where $\mathbf{1}$ is a vector of dimension equal to the number of transitions having all elements equal to one, while $\boldsymbol{V}$ is a matrix, each row corresponding to a characteristic vector of an observation, i.e., the first row is $\boldsymbol{v}_1$ (the characteristic vector of π_1), the second row is $\boldsymbol{v}_2$ etc.

Using a $\boldsymbol{\sigma}$ solution of (4), the sequence of regions that should be crossed by the robots can be easily obtained. As we mentioned, state equation is not a necessary and sufficient condition for the reachability of a marking but, in this case, the RMPN belongs to the class of state-machines Petri nets and in order to fulfill the task it is not necessary to perform cycles. In this case, we show in (Mahulea and Kloetzer, 2018) that the sequence of transitions can be easily found from the vector $\boldsymbol{\sigma}$.

Example 2. Let us consider the environment in Fig. 1(a) containing three regions of interest (p_7, p_9 and p_{11}) and two robots initially located in p_8 (robot r_1) and p_6 (robot r_2). The RMPN system is shown in Fig. 1(b) and it is composed by a set of 12 places, $P = \{p_1, p_2, \ldots, p_{12}\}$ and a set of 28 transitions, $T = \{t_{11,12}, t_{12,11}, \ldots\}$. The set of output symbols is $\Pi = \{\pi_1, \pi_2, \pi_3\}$ while the observation function is $h(p_{11}) = \pi_1$; $h(p_9) = \pi_2$; $h(p_7) = \pi_3$ and $h(p_i) = \emptyset$ for all $p_i \in P \setminus \{p_7, p_9, p_{11}\}$. The initial marking $\boldsymbol{m}_0$ is a vector having all elements equal to zero except $m_0[p_6] = m_0[p_8] = 1$, since there is a robot initially located in p_6 and another one in p_8.

Assume that the team should fulfill the following Boolean formula at the final state: $\varphi = \pi_1 \wedge \pi_2 \wedge \neg\pi_3$, meaning that a robot should stop in p_{11} (since $h(p_{11}) = \pi_1$), one robot should stop in p_9 (since $h_9 = \pi_2$) and no robot should be in p_7 (since $h(p_7) = \pi_3$).

By solving the MILP (4), the following plan is obtained for each robot:

- robot 1: $p_8 p_3 p_1 p_{12} p_{11}$,
- robot 2: $p_6 p_{10} p_5 p_9$.

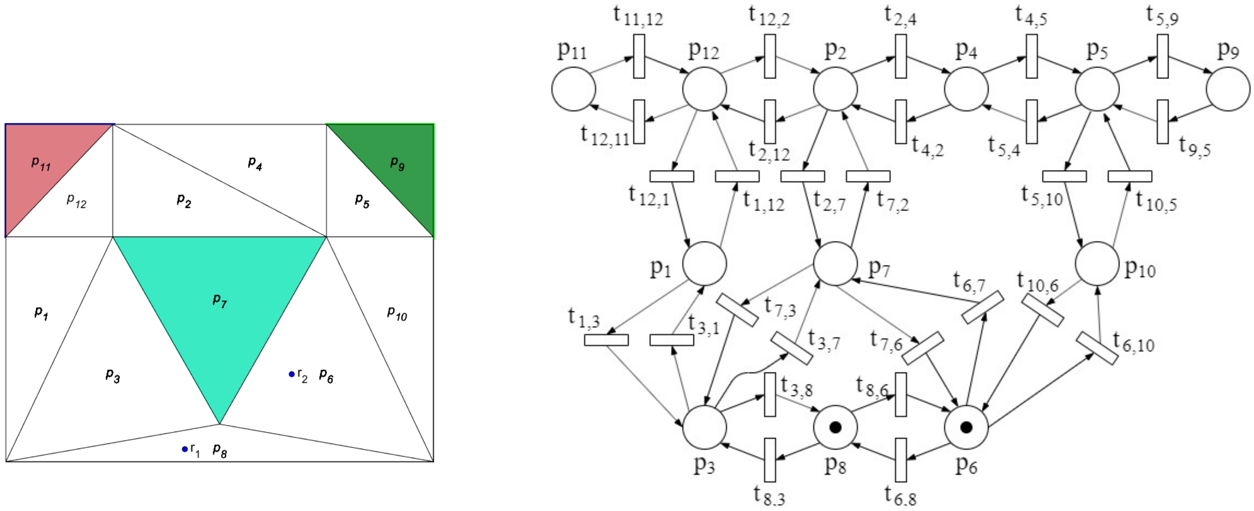
However, the solution has been obtained assuming that the environment map is known by the robots. In the following section we provide an approach for the case in which the robots have only partial knowledge of the environments and each one is separately computing an individual plan by locally solving an instance of problem (4) based on its current belief on the locations of regions of interest.

4. DISTRIBUTED REGION ESTIMATION

Let us assume now that the robots do not know a priory the labeling of the different regions, p_i, but they are able to obtain noisy observations, $o_r(p_i) \in \Pi, r \in R$, depending on the region where they are. Let $\mathbb{P}(\pi_k)$, be the probability of a given region $p \in P$ being labeled as $\pi_k \in \Pi$. Then, given a set of observations, $o_r(p_i)$, the objective of the team is to obtain a probability distribution, $\mathbb{P}(\Pi)$, taking into account the limited communications of the network.

In order to do this, we combine the distributed inference algorithm described in Julian et al. (2012) together with the fast consensus iteration of Montijano et al. (2013). For simplicity, we will describe the algorithm for a single region, $p \in P$, noting that the same algorithm will be replicated for all the regions in the map.

First of all, assume that the robots know the calibration of their sensors, so that they know the likely-hood of obtaining a given observation given the label, $\mathbb{P}(\pi_{k1}|\pi_{k2})$. For example, if the robot is located in region p, we already know that $\mathbb{P}(h(p)|h(p)) = 1$ whereas it will be zero for any other $\pi_k \neq h(p)$. On the other hand, if p is far away from the robot, we assume that $\mathbb{P}(\pi_k|h(p)) = 1/|\Pi|$ for all π_k, which basically means that the robot can measure every label with the same probability.



(a) An environment.

(b) Corresponding RMPN.

Fig. 1. Example of an environment and the Petri net model

Let $\mathbb{P}_t^{r_i}(\Pi)$, be the prior distribution that robot r_i has of the region at planning round t. Applying Bayes rule for a single robot, given the observation $o_{r_i}(p)$, the posterior is

$$\mathbb{P}_t^{r_i}(\pi_k|o_{r_i}(p)) = \frac{\mathbb{P}_t^{r_i}(\pi_k)\mathbb{P}(o_{r_i}(p)|\pi_k)}{\sum_{k\in|\Pi|}\mathbb{P}_t^{r_i}(\pi_k)\mathbb{P}(o_{r_i}(p)|\pi_k)}. \quad (5)$$

Recalling that we assume that the labeling of the regions does not change over time, the prior at the next planning time is simply $\mathbb{P}_{t+1}^{r_i}(\Pi) = \mathbb{P}_t^{r_i}(\Pi|o_{r_i}(p))$.

Assuming no correlation between the observations of the different robots, Julian et al. (2012) proposes to approximate the combined likely-hood, $\mathbb{P}(O(p)|\pi_k)$ by the product,

$$\mathbb{P}(O(p)|\pi_k) = \prod_{r\in|R|}\mathbb{P}(o_r(p)|\pi_k). \quad (6)$$

This quantity can be obtained in a distributed manner by means of a consensus iteration, which needs to be run until convergence. Since this iteration can require many communication rounds to converge, in this paper we exploit the polynomial solution presented in Montijano et al. (2013) to accelerate the process. Let $x_i(0) = \log(\mathbb{P}_t^{r_i}(o_{r_i}(p)|\pi_k))$ be the initialization that robot r_i makes to estimate (6). Then, the following distributed algorithm

$$x_i(1) = \frac{1}{T_1(c)}\sum_{j\in\mathcal{N}_i\cup i} w'_{ij}x_j(0), \quad (7a)$$

$$x_i(n) = 2\frac{T_{n-1}(c)}{T_n(c)}\sum_{j\in\mathcal{N}_i\cup i} w'_{ij}x_j(n-1) - \frac{T_{n-2}(c)}{T_n(c)}x_i(n-2), \quad (7b)$$

is used to estimate the average of the initial conditions of the whole set of robots. In the algorithm, $T_n(c)$ denotes the Chebyshev polynomial of the first kind of degree n, computed recursively by $T_n(c) = 2T_{n-1}(c)c - T_{n-2}(c)$, the parameter $c > 1$ is a design parameter and $0 < w_{ij} < c$ is the weight associated to the exchange between neighboring robots r_i and r_j, denoted by $\mathcal{N}_i$. This matrix needs to be balanced with row-sum equal to c for all r_i. Finally, the value in (6) is obtained locally by

$$\mathbb{P}(O(p)|\pi_k) = \frac{\exp(x_i(n))^{|R|}}{\sum\exp(x_i(n))^{|R|}}. \quad (8)$$

5. COMBINED ALGORITHM

Let us now describe how the planning and estimation algorithms work together. Initially, the environment partition P is available to each robot, and the formula φ is read and converted to a set of linear inequalities as (2). Based on P, each robot r_i constructs a RMPN model as in Def. 1, except the observation function h. This model will be iteratively adjusted after each robot movement, as follows.

The initial marking is chosen based on the known actual position of robot r_i and on the estimate that r_i has on the other robot's current positions. For constructing the observation function h, robot r_i estimates the probability of each cell from P belonging to a region from Π. To this end, r_i communicates with all the robots in its communication range and adjusts its estimations as in Sec. 4. After that, r_i chooses a deterministic output function h by simply considering the maximum probability of observing a region from Π in each place, $h(p) = \max_k \mathbb{P}_t^{r_i}(\pi_k)$, and then it constructs the matrix V containing the characteristic vectors of observations. Having this information, robot r_i formulates and solves MILP (4).

If the optimization problem does not have a feasible solution, the robot will randomly choose an adjacent partition region to move. Note that it is possible that MILP (4) does not have a solution, because of the lack of any information on some regions from Π, which triggers some null characteristic vectors. If problem (4) returns a solution, r_i chooses the next region to move based on its first transition in RMPN yielded by solution $\boldsymbol{\sigma}$. Now, robot r_i executes a movement in the partition and then it reiterates the whole procedure starting from the adjustment of RMPN model.

Note that each robot movement is either enforced by solution of MILP (4), or it is randomly chosen when the optimization problem is infeasible. Of course, the "movement" includes the possibility of remaining in the

Algorithm 1. Iterative construction of solution for robot r_i

Input: $P = \{p_1, p_2, \ldots, p_{|P|}\}$ the set of regions composing the environment, $\Pi = \{\pi_1, \pi_2, \ldots, \pi_\Pi\}$ the set of atomic propositions, $h : P \to \{\Pi \cup \{\emptyset\}\}$ labelling function, $R = \{r_1, r_2, \ldots, r_{|R|}\}$ the set of robots and their initial region in P; $RADIUS$ - the communication radius; MAX_ITER - maximum number of iterations

Output: Sequence of cells traversed by the robot r_i.

Construct the RMPN system $\mathcal{N}_i = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}, \Pi, h, \boldsymbol{m}_0\rangle$;

Read the formula φ and compute the corresponding set of linear inequalities $\boldsymbol{A}_{task} \cdot \boldsymbol{x} \leq \boldsymbol{b}_{task}$;

while $num\_iteration < MAX\_ITER$ **do**

- Acquire a measurement of each cell of the map;
- Communicate with all robots $r_j \in R$, $r_j \neq r_i$ located within the communication radius $RADIUS$ and compute common estimations;
- Update the initial marking $\boldsymbol{m}_0$ of r_i the position of the robots with which r_i has been communicated;
- Compute matrix V (characteristic matrix observation) based on a deterministic output function h assuming maximum probability in the robot estimation;
- Solve MILP (4);
- **if** *MILP* (4) *is infeasible* **then**
  - Chose a random move from the adjacent cells including the one where the robot is;
  - Update $\boldsymbol{m}_0$ assuming the other robots are not moving;
- **else**
  - Advance to the next cell according to the solution of MILP (4);
  - Update $\boldsymbol{m}_0$ assuming that the other robots are moving according to the solution of MILP (4);
- Wait until the maximum time of a step elapses.

current place, if MILP solution yields such a transition or the random decision chooses the current partition cell. All robots should complete their movements before exchanging information within their communication range and making new estimates on positions of regions Π (outputs of RMPN). For this, after its movement, each robot is required to wait such that all robots completed their movements, and this can be accounted by imposing a waiting time based on the maximum time required for moving between any two adjacent cells from partition P.

The number of iterations where each robot estimates the outputs and solves a MILP is upper-bounded by a user-chosen value. The larger this value, the greater the chance that all robots reach stopping positions where the team satisfies the Boolean formula φ. Alg. 1 summarizes the steps from this section, and this algorithm is run by every robot r_i.

6. SIMULATIONS

Let us consider first the environment in Ex. 2. By applying Alg. 1, in 13 iterations both robots stop in the final regions p_9 and p_{11}. Robot r_1 does the following movements: p_8, p_3, p_7, p_3, p_1, p_3, p_1, p_3, p_8, p_6, p_{10}, p_5 and then stops in p_9. Robot r_2 performs the following movements p_6, p_{10}, p_5, p_4, p_2, p_{12}, p_{11} and remains in p_{11}. Fig. 2 illustrates some belief maps of the robots along their runs, i.e. they show where each robot thinks the regions from Π are located. Notice that before the forth iteration, the belief map of the second robot does not contain a "red region" and robot r_2 performs random moves, because its instance of MILP (4) is infeasible.

The second simulation considers a more challenging environment composed by 400 regions. There are three regions of interest, e.g., $|\Pi| = 3$, 9 robots in the team and the Boolean formula is given by $\varphi = \pi_1 \wedge \pi_2$. By using a Petri Nets model, we are able to solve the 9 instances of the MILP required at each planning iteration in approximately 0.4 seconds using a standard desktop computer (i7 at 3.4GHz with 32Gb of RAM). The trajectories followed by the robots are given in Fig. 3, where we can observe that in the end the Boolean formula is satisfied.

7. CONCLUSION

This paper has proposed a novel algorithmic solution for planning a path for a distributed multirobot system in an unknown environment. The robots should reach final states such that the team satisfies an imposed Boolean formula over a set of regions of interest that exist in the environment, but whose positions are not initially known. The environment is partitioned, while the robots can noisily sense a surrounding area and can communicate in a certain radius. The algorithm builds on a planning method based on Boolean formulas and RMPN models in known environments and on estimation and communication techniques for agents that have probabilistic information of the regions of interest. The developed solution consists in an algorithm that is iterated by each individual robot. At each iteration step, the robot updates its information on the map, solves a MILP optimization problem and takes a movement to a neighboring partition region. For the steps when the optimization problem is infeasible, the robot randomly moves in order to improve its environment estimation. Future work will focus on changing these random movements with guided strategies whenever the current MILP is infeasible, and on investigating the influence of a few parameters on the number of iterations needed for fulfilling the specification.

REFERENCES

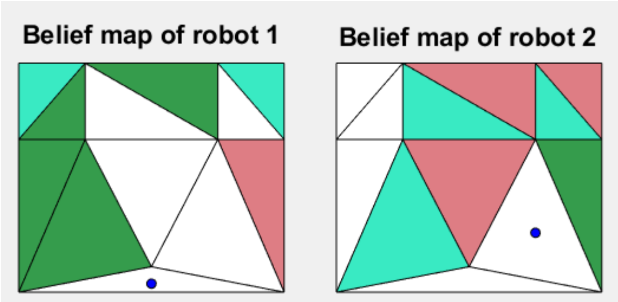
Belta, C., Bicchi, A., Egerstedt, M., Frazzoli, E., Klavins, E., and Pappas, G. (2007). Symbolic Planning and Control of Robot Motion. *IEEE Robotics and Automation Magazine*, 14(1), 61 – 71.

Berg, M.D., Cheong, O., and van Kreveld, M. (2008). *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition.

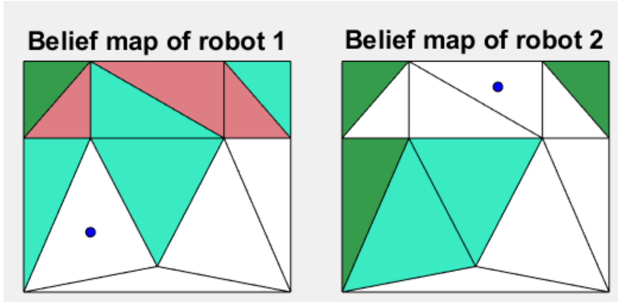
Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E., and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Boston.

Ding, X., Smith, S.L., Belta, C., and Rus, D. (2014). Optimal control of Markov decision processes with linear temporal logic constraints. *IEEE Transactions on Automatic Control*, 59(5), 1244–1257.

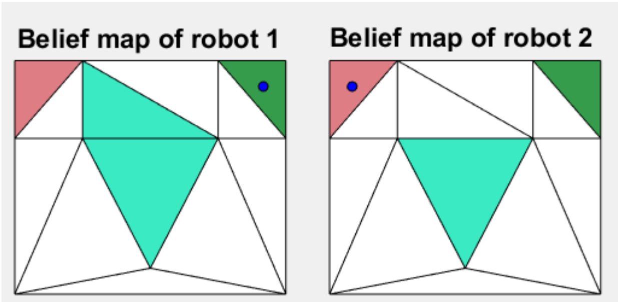
Fainekos, G.E., Girard, A., Kress-Gazit, H., and Pappas, G.J. (2009). Temporal logic motion planning for dynamic robots. *Automatica*, 45(2), 343–352.



Belief maps of the robots before first movement.



Belief maps of the robots before the 4^{th} iteration.



Belief maps when the robots stop.

Fig. 2. Belief maps after different iterations of Alg. 1 for the scenario in Ex. 2.

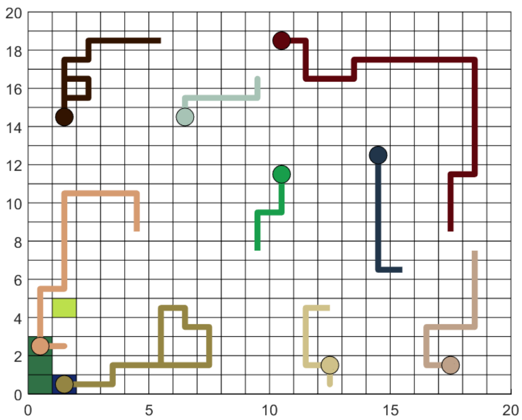


Fig. 3. Trajectories of the 9 robots. The dots indicate the final position of each robot. It can be seen at the bottom left of the figure that there are two robots in the two colored regions, which implies the satisfaction of the Boolean formula.

Garrido, S., Moreno, L., Blanco, D., and Martín-Monar, F. (2010). Robotic motion using harmonic functions and finite elements. *J. Intell. Robotics Syst.*, 59(1), 57–73.

Guo, M., Tumova, J., and Dimarogonas, D. (2014). Cooperative decentralized multi-agent control under local ltl tasks and connectivity constraints. In *IEEE Conf. on Decision and Control*, 75 – 80. Los Angeles, USA.

Huang, G.Q., Rad, A.B., and Wong, Y.K. (2005). Online slam in dynamic environments. In *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics*, 262–267.

Julian, B.J., Angermann, M., Schwager, M., and Rus, D. (2012). Distributed robotic sensor networks: An information-theoretic approach. *The International Journal of Robotics Research*, 31(10), 1134–1154.

Kavraki, L., Svestka, P., Latombe, J., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), 566 – 580.

Kloetzer, M. and Mahulea, C. (2015). LTL-Based Planning in Environments With Probabilistic Observations. *IEEE Transactions on Automation Science and Engineering*, 12(4), 1407–1420.

Kloetzer, M. and Mahulea, C. (2020). Path planning for robotic teams based on LTL specifications and Petri net models. *Discrete Event Dynamic Systems: Theory and Applications*, 30, 55–79.

Latombe, J.C. (1991). *Robot Motion Planning*. Kluger Academic Pub.

LaValle, S.M. (2006). *Planning Algorithms*. Cambridge. Available at `http://planning.cs.uiuc.edu`.

Mahulea, C. and Kloetzer, M. (2018). Robot Planning based on Boolean Specifications using Petri Net Models. *IEEE Trans. on Automatic Control*, 63(7), 2218–2225.

Mahulea, C., Kloetzer, M., and González, R. (2020). *Path Planning of Cooperative Mobile Robots Using Discrete Event Models*. Wiley-IEEE Press.

Montijano, E., Montijano, J.I., and Sagues, C. (2013). Chebyshev polynomials in distributed consensus applications. *IEEE Trans. on Signal Processing*, 61(3), 693–706.

Svorenova, M., Tumova, J., Barnat, J., and Cerna, I. (2012). Attraction-based receding horizon path planning with temporal logic constraints. In *IEEE Conference on Decision and Control*, 6749–6754.

Zamora, E. and Yu, W. (2014). Recent advances on simultaneous localization and mapping for mobile robots. *IETE Technical Review*, 30(6), 490–496.