DOI: 10.48011/asba.v2i1.1593

# Robust Connectivity for Multiagent Systems using Mixed-Integer Programming ⋆

**Angelo Caregnato-Neto** * **Marcos R. O. A. Maximo** ** **Rubens J. M. Afonso** †,*

* *Electronic Engineering Division, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, (e-mail: a.caregnato@gmail.com),(e-mail: rubensjm@ita.br).*
** *Autonomous Computational Systems Lab (LAB-SCA), Computer Science Division, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, (e-mail: mmaximo@ita.br).*
† *Institute of Flight System Dynamics, Technical University of Munich, Bayern, Germany (e-mail: rubens.afonso@tum.de).*

**Abstract:** This work is concerned with the problem of planning trajectories for a group of mobile robots operating in a obstacle-filled environment. The objective of the robots is to cooperatively reach targets sets, avoiding collisions between themselves and with obstacles, while maintaining the robust connectivity of their communication network. Model predictive control with mixed-integer encoding is used to plan the trajectories in a centralized fashion. Two novel constraints are introduced to cope with the robustness connectivity problem. Simulations are provided to evaluate the proposal.

*Keywords:* Multiagent system, Trajectory planning, Robust connectivity, Mixed-integer programming, Model predictive control.

## 1. INTRODUCTION

Multiagent systems (MAS) are systems of multiple agents that collaborate to solve tasks (Dorri et al., 2018). In particular, MAS composed of autonomous mobile agents (mobile robots) recently became viable due to improvements in their communication capability, as well as decreased fabrication costs and better autonomy of the robots. A group of coordinated robots can succeed in situations where a single robot is insufficient; for example, complex and large-scale problems such as load transportation (Klausen et al., 2018) and forest firefighting (Harikumar et al., 2018).

One of the main challenges regarding this type of MAS is trajectory planning (Garcia et al., 2013). Multiple trajectories must be planned such that collisions with obstacles and between robots are avoided. Moreover, the robots must be guided so as to cooperatively work towards the same global goal, which is often achieved by allocating them to different tasks. Many of these requirements are reliant in the constant communication between robots. Communication networks of MAS are commonly modelled as graphs. Therefore, to achieve constant communication between robots, the graph must be connected at all times.

In Filotheou et al. (2018) the problems of collision avoidance and connectivity of MAS were addressed using model predictive control (MPC). However, the task allocation was pre-defined. Sun et al. (2018) proposed a sliding mode control law to guarantee the robustness of the connectivity network of a MAS subjected to disturbances, but a scenario without obstacles was considered. In Afonso et al. (2020), the problems of collision and obstacle avoidance, as well as connectivity and task allocation are solved within a MPC with mixed-integer linear programming (MILP) encoding. Nevertheless, this work does not address the robustness of the communication network. Therefore, the failure of a single robot could compromise the operation of the group.

The present work contributes by expanding the results of Afonso et al. (2020) by proposing novel connectivity constraints which guarantee a degree of robustness to the communication network of the MAS. The subsequent sections are organized as follows. Section 2 details the problem. In Section 3 the solution of the problem is presented. In Section 4 the simulations results are displayed and discussed. Finally, Section 5 concludes the paper and shares our ideas for future work.

## 2. PROBLEM STATEMENT

Consider a group of $N_a$ robots. The dynamics of these robots are modelled by the following discrete-time equation

$$\mathbf{x}_{i,k+1} = \mathbf{A}\mathbf{x}_{i,k} + \mathbf{B}\mathbf{u}_{i,k},\ 1 < i \le N_a, \tag{1}$$
$$\mathbf{y}_{i,k} = \mathbf{C}\mathbf{x}_{i,k},\ 1 \le i \le N_a, \tag{2}$$

where $\mathbf{x}_{i,k} \in \mathbb{R}^{n_x}$ is the state vector, $\mathbf{u}_{i,k} \in \mathbb{R}^{n_u}$ is the control vector, and $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$ are state space matrices; $\mathbf{y}_{i,k} \in \mathbb{R}^{n_y}$ is the vector of

⋆ Ângelo Caregnato-Neto acknowledges the support of CAPES for his scholarship. Rubens Afonso acknowledges CAPES (grant 88881.145490/2017-01) and the German Ministry of Education and Research through the Alexander von Humboldt foundation for his fellowship.

positions, and $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$ is a matrix that extracts the positions from the state vector. Additionally, let $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{U} \subset \mathbb{R}^{n_u}$ denote compact polytopic sets representing admissible values for the states and controls, respectively.

For simplicity, let us consider that the group operates within a 2-dimensional region, represented by a compact polygonal set $\mathcal{A} \subseteq \mathbb{R}^2$, i.e. $n_y = 2$. Consider that this region contains $N_t$ target sets $\mathcal{Q}_p \subset \mathbb{R}^2$, $1 \leq p \leq N_t$, which are also compact and polygonal. These sets represent advantageous or required areas to visit during the operation of the group. The region $\mathcal{A}$ also contains $N_o$ static polygonal obstacles, denoted by $\mathcal{O}_q \subset \mathbb{R}^2$, $1 \leq q \leq N_o$. The area occupied by each robot, at some time-step $k$, is represented by the compact polygonal sets $\mathcal{R}_{i,k} \subset \mathbb{R}^2$, $1 \leq i \leq N_a$. The free operation regions for each robot, at a time-step $k$, are then defined as:

$$\mathcal{A}_{i,k}^{\text{free}} \triangleq \mathcal{A} \backslash \left\{ \bigcup_{q=1}^{N_o} \mathcal{O}_q \;\cup\; \bigcup_{\substack{j=1 \\ j \neq i}}^{N_a} \mathcal{R}_{j,k} \right\}, 1 \leq i \leq N_a. \tag{3}$$

The robots are able to communicate if they are within each other's communication areas, which are modelled as compact polygonal sets $\mathcal{C}_{i,k} \subset \mathbb{R}^2$, $1 \leq i \leq N_a$. All regions are assumed to have the same size, therefore the communication connections are always bidirectional. The communication network is represented by a time-varying undirected graph $\mathcal{G}_k = \{\mathcal{V}, \mathcal{E}_k\}$, where $\mathcal{V} = \{v_1, \ldots, v_{N_a}\}$ is a set of vertices, and $\mathcal{E}_k \subseteq \{(v_i, v_j) \mid v_i, v_j \in \mathcal{V}\}$ a set of edges. The vertices represent the robots, whereas the edges represent possible communication links between them.

In this context, the problem is to plan trajectories for each robot, such that the following requirements are met:

i) $\mathcal{G}_k$ is connected with a degree of robustness;

ii) control and state limits are respected;

iii) no collisions between robots or with obstacles occur;

iv) the maneuver ends upon visitation of the last target $\mathcal{Q}_{N_t}$;

v) the robots cooperate to reach a compromise between target set visitation, fuel consumption, and time expenditure.

The particular contribution of this paper is related to requirement i).

## 3. PROBLEM SOLUTION

In this section, we introduce a formulation of MPC used to calculate trajectories compatible with the aforementioned requirements. First, an encoding addressing requirements ii), iii), iv), and v) is presented. Then, two solutions for the robust connectivity problem are proposed. These approaches require the introduction of new constraints to the optimization problem.

### *3.1 MAS Trajectory Planning with MILP*

In order to cope with requirements ii) to v), the optimization problem proposed in Afonso et al. (2020) is utilized.

**Problem 1.**

$$\min_{\mathbf{u}_{i,k}, b_{p,k}, N} N + \gamma \sum_{k=0}^{N} \sum_{i=1}^{N_a} \|\mathbf{u}_{i,k}\|_1 - R \sum_{k=0}^{N} \sum_{p=1}^{N_t} b_{p,k}, \tag{4}$$

subject to

$$\mathbf{x}_{i,0} = \mathbf{x}_i^{\text{initial}},\ 1 \leq i \leq N_a, \tag{5a}$$
$$\mathbf{x}_{i,k+1} = \mathbf{A}\mathbf{x}_{i,k} + \mathbf{B}\mathbf{u}_{i,k},\ 1 \leq i \leq N_a,\ 0 \leq k \leq N, \tag{5b}$$
$$\mathbf{x}_{i,k} \in \mathcal{X},\ 1 \leq i \leq N_a,\ 0 \leq k \leq N+1, \tag{5c}$$
$$\mathbf{u}_{i,k} \in \mathcal{U},\ 1 \leq i \leq N_a,\ 0 \leq k \leq N, \tag{5d}$$
$$\mathbf{y}_{i,k} \in \mathcal{A}_{i,k}^{\text{free}},\ 1 \leq i \leq N_a,\ 0 \leq k \leq N+1, \tag{5e}$$
$$\exists i \mid \mathbf{y}_{i,N+1} \in \mathcal{Q}_{N_t}, \tag{5f}$$
$$b_{i,p,k} = 1 \Rightarrow\ \mathbf{y}_{i,k} \in \mathcal{Q}_p,\ 1 \leq p \leq N_t,\ 0 \leq k \leq N+1,\ 1 \leq i \leq N_a, \tag{5g}$$
$$\sum_{i=1}^{N_a} \sum_{k=0}^{N} b_{i,p,k} \leq 1,\ 1 \leq p \leq N_t. \tag{5h}$$

Constraints (5c) and (5d) impose limits on the state and control values, therefore requirement ii) is satisfied. Constraint (5e) addresses the collision avoidance both among robots themselves and between robots and obstacles, i.e., requirement iii). Without loss of generality, constraint (5f) guarantees requirement iv) by imposing that the last target set is reached at step $N+1$. Requirement v) is achieved by balancing the weights $\gamma \in \mathbb{R}_+$ and $R \in \mathbb{R}_+$ in the cost function depicted in Equation (4). The minimization of time is achieved through the use of a variable horizon $N \in \mathbb{N}$, which is an optimization variable. Constraints (5g) impose that a reward is only granted if a target set is visited by a robot; constraint (5h) ensures that the reward is collected once per target set.

Constraint (5g) depicts the explicit use of binary variables in this problem; these variables are naturally implemented in a MPC-MILP problem. Additionally, the imposition of constraints such as (5e) results in the non-convexity of the feasible region. This problem is addressed with the "big-M" technique, which is implemented with binary variables (Agarwal et al., 2010). We refer the reader to Afonso et al. (2020) for details regarding the implementation of this optimization in a linear mixed-integer programming framework.

*Remark 1.* The variable horizon $N$ is selected from a set $\{1, \ldots, T\}$, where $T$ is the fixed horizon of the optimization problem. Since the value of $N$ is unknown *a priori*, all constraints are implemented for $0 \leq k \leq T$. However, they are relaxed once the variable horizon is reached. This also results in $\mathbf{u}_{i,k} = 0,\ 1 \leq i \leq N_a,\ N+1 \leq k \leq T$.

### *3.2 Robust Communication Network*

Let $\mathcal{G}_k$ be a graph that represents the communication network of a MAS. In this work, a network is said to be robust if, at some time-step $k$, $\mathcal{G}_k$ remains connected despite the removal of a single vertex; such a scenario is consistent with the full failure of a robot, or the failure of its communication hardware. As connectivity is not lost instantly by the removal of a single robot from the network, the robots can rearrange to form a robust network again once a fault is identified. In the context of the present work, this property is enough to satisfy requirement i).

First, we introduce some concepts used in the analysis of the connectivity of graphs.

*Definition 1.* A *cut vertex* of a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is any $v \in \mathcal{V}$ such that $\mathcal{G}' = \{\mathcal{V}/\{v\}, \mathcal{E}'\}$, $\mathcal{E}' \triangleq \mathcal{E} \backslash \{(v, v_i) \mid \forall v_i \in \mathcal{V}\}$, is disconnected (Bertrand and Zhang, 2012). The number of cut vertices of a graph $\mathcal{G}$ is denoted by $\delta(\mathcal{G}) \in \mathbb{N}$.

*Definition 2.* A *vertex-cut* in a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is any set of vertices $\mathcal{K} \subseteq \mathcal{V}$, such that $\mathcal{G}' = \{\mathcal{V}/\mathcal{K}, \mathcal{E}'\}$, $\mathcal{E}' \triangleq \mathcal{E} \backslash \{(v_i, v_j) \mid \forall v_i \in \mathcal{K}, \forall v_j \in \mathcal{V}\}$, is disconnected. A minimum vertex-cut $\mathcal{K}^{min}$ is a vertex-cut with minimum cardinality (Bertrand and Zhang, 2012).

*Definition 3.* The *connectivity* $\kappa$ of a graph $\mathcal{G}$, is the cardinality of the minimum vertex-cut (Bertrand and Zhang, 2012).

*Definition 4.* A graph $\mathcal{G}$ is said to be *k-connected* if $\kappa(\mathcal{G}) \geq k$ (Bertrand and Zhang, 2012).

*Definition 5.* A *walk* is a sequence of adjacent vertices of a graph. A *closed trail* is a walk where the first and last vertices of the sequence are equal and each edge is traversed only once. A *cycle* is a closed trail of length three or more with no repeated vertices. (Bertrand and Zhang, 2012).

According to Definitions 1, 2, and 3, if the communication network of the MAS is represented by a *2-connected* graph, then it possesses the discussed degree of robustness, i.e., the network (graph) would remain connected despite the removal of a single robot (vertex). Indeed, such a graph has connectivity $\kappa(\mathcal{G}) \geq 2$, therefore $|\mathcal{K}^{min}| \geq 2$. From Definition 1, it follows that at least two vertices must be removed to disconnect $\mathcal{G}$.

We now discuss a class of graphs denoted *Hamiltonian*, which are always 2-connected (Bertrand and Zhang, 2012).

*Definition 6.* A cycle in a graph $\mathcal{G}$ is called a *Hamiltonian cycle* if it contains every vertex of $\mathcal{G}$. A graph that contains a Hamiltonian cycle is called *Hamiltonian graph* (Bertrand and Zhang, 2012).

As an illustration, consider the example presented in Figure 1. In a Hamiltonian graph, there are always two paths (through the Hamiltonian cycle) to any vertex: clockwise and counter-clockwise. Therefore, the removal of a single vertex disrupts only one of these paths, and the remaining path still connects the vertices.

*Remark 2.* The hamiltonicity of a graph is a sufficient condition for 2-connectivity.

Finally, sufficient conditions for the hamiltonicity of a graph are presented.

*Theorem 1.* Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a graph with $n \geq 3$ vertices. If $deg(u) + deg(v) \geq n$, for all pairs of nonadjacent vertices $u, v \in \mathcal{V}$, then $\mathcal{G}$ is Hamiltonian (Ore, 1960).

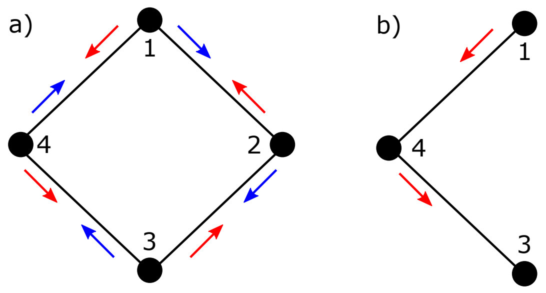


Figure 1. (a) Example of a Hamiltonian graph and (b) resulting graph if vertex 2 is removed.

*Corollary 1.* Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a graph with $n \geq 3$ vertices. If $deg(v) \geq n/2$, $\forall v \in \mathcal{V}$, then $\mathcal{G}$ is Hamiltonian (Bertrand and Zhang, 2012).

In the following subsection, the results in Theorem 1 and Corollary 1 will be used to encode alternative MILP constraints that ensure 2-connectivity of the communication graph of the MAS.

### *3.3 Mixed-Integer Programming for Robust Connectivity*

First, the connections between robots are encoded through the following constraints

$$b^{con}_{i,j,k} = 1 \Rightarrow \mathbf{y}_{i,k} \in \mathcal{C}_{j,k}, \ 1 \leq i, j \leq N_a, \tag{6}$$

for $0 \leq k \leq N$. Therefore, the $i$th and $j$th robots are connected, i.e., within each other's communication regions, at a time-step $k$, if $b^{con}_{i,j,k} = 1$. The number of connections of a robot is equal to the degree of its corresponding vertex in the graph that describes the communication network. Let $deg_{i,k}$ denote the degree of the $i$th vertex at a time-step $k$, then this quantity is associated with the binary variables through the following equation:

$$deg_{i,k} = \sum_{j=1}^{i-1} b^{con}_{j,i,k} + \sum_{j=i+1}^{N_a} b^{con}_{i,j,k}, \ 1 \leq i \leq N_a, \ 0 \leq k \leq N. \tag{7}$$

An example is now provided to illustrate the use of Equation (7).

**Example 1.** Consider a bidirectional graph $\mathcal{G}_k$ with 4 vertices and the following adjacency matrix

$$Adj = \begin{bmatrix} 0 & 1 & 0 & 0 \\ * & 0 & 1 & 1 \\ * & * & 0 & 0 \\ * & * & * & 0 \end{bmatrix}, \tag{8}$$

where * represents values that do not matter. Let us calculate the degree of the second vertex. This can be accomplished by adding the sum of all elements of the second row with the sum of all elements of the second column; therefore, the degree of the second vertex is 3. Assume now that $\mathcal{G}_k$ represents a communication network in our scheme, then an analogous operation is performed using Equation (7)

$$deg_{2,k} = \underbrace{\left(b^{con}_{2,3,k} + b^{con}_{2,4,k}\right)}_{row} + \underbrace{b^{con}_{1,2,k}}_{column} = 3. \tag{9}$$

Two robust connectivity constraints are now proposed; each one guarantees a robust communication network for the MAS. As in Theorem 1, constraint (10) imposes a minimum value for the sum of the degree of each nonadjacent vertex:

$$deg_{i,k} + deg_{\ell,k} \geq N_a - N_a b^{con}_{i,\ell,k},\ 1 \leq i \leq N_a - 1, \quad i+1 \leq \ell \leq N_a. \tag{10}$$

Notice that, if the $i$th and $j$th robots are connected at a time step $k$, i.e., their corresponding vertices in the graph are adjacent, then $b_{i,j,k} = 1$ and the constraint is relaxed, since the degrees of vertices are always nonnegative. The number of constraints in (10) is $(T+1)N_a(N_a-1)/2$, which is $O(TN_a^2)$.

As in Corollary 1, constraint (11) imposes a minimum value for the degree of each vertex:

$$deg_{i,k} \geq \frac{N_a}{2},\ 1 \leq i \leq N_a,\ 0 \leq k \leq N. \tag{11}$$

The number of constraint in this case is $(T+1)N_a = O(TN_a)$. Notice that constraint (10) scales quadratically with the number of robots in the group, whereas constraint (11) scales linearly. Consequently, as the number of robots in the group increases, an encoding with the second constraint scales significantly better than an encoding with the first constraint. On the other hand, since the condition in Corollary 1 is sufficient for the condition in Theorem 1, it follows that constraint (11) is more conservative than constraint (10).

*Remark 3.* The dominant constraint in terms of computational complexity is the one associated with collision avoidance, with $O(TN_a^2 n)$, where $n \in \mathbb{N}$ is the number of sides of the polytope representing the collision region.

Constraints (10) and (11) provide a degree of connectivity robustness which is absent in the formulations presented in Afonso et al. (2020), where only 1-connectivity is guaranteed. However, our approach is based on sufficient conditions, whereas the two constraints proposed in Afonso et al. (2020) are associated with necessary and necessary sufficient conditions; these constraints scale linearly and exponentially, respectively.

## 4. SIMULATIONS

In order to evaluate the proposed solutions, a simulation is carried out. Our methodology is to compare the results of Problem (1) with the addition of constraint (6) and three different connectivity constraints: constraint (10), constraint (11), and the faster yet more conservative approach presented in Afonso et al. (2020). The results are compared in terms of performance and robustness through the resulting optimal cost and average number of cut-vertices

$$\bar{\delta} \triangleq \frac{1}{N} \sum_{k=0}^{N} \delta\left(\mathcal{G}_k\right), \tag{12}$$

respectively.

### *4.1 Simulation details*

The robots are assumed to be satisfactorily modelled as double integrators. The discrete-time state matrices for a sample time of 1 second, are

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 0.5 & 1 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{bmatrix}. \tag{13}$$

The state vector is defined as $\mathbf{x}_{i,k} \triangleq \left[r^x_{i,k}, v^x_{i,k}, r^y_{i,k}, v^y_{i,k}\right]^T$, with $r^x_{i,k} \in \mathbb{R}$ and $r^y_{i,k} \in \mathbb{R}$ being the position coordinates, whereas $v^x_{i,k} \in \mathbb{R}$ and $v^y_{i,k} \in \mathbb{R}$ are the corresponding velocities. The control vector $\mathbf{u}_{i,k} \triangleq \left[a^x_{i,k}, a^y_{i,k}\right]^T$ is composed of the components of acceleration $a^x_{i,k} \in \mathbb{R}$ and $a^y_{i,k} \in \mathbb{R}$.

A scenario with $N_a = 6$, $N_o = 2$, and $N_t = 5$ is considered. Table 1 details each polygonal region of the scenario in terms of type, center, and side lengths. The fixed horizon size is chosen as $T = 20$; the weights are selected as $\gamma = 0.1$ and $R = 10$.

Table 1. Parameters of the polygonal regions in the scenario.

| Region | Type | Center | Sides length |
|---|---|---|---|
| $\mathcal{A}$ | Square | $(5, 5)$ | 10 |
| $\mathcal{O}^1$ | Rectangle | $(1.622, 3.113)$ | $(1.663, 1.025)$ |
| $\mathcal{O}^2$ | Rectangle | $(7.943, 5.285)$ | $(3.408, 1.808)$ |
| $\mathcal{Q}^1$ | | $(1.683, 2.169)$ | |
| $\mathcal{Q}^2$ | | $(1.107, 8.013)$ | |
| $\mathcal{Q}^3$ | Square | $(9.406, 4.034)$ | 0.5 |
| $\mathcal{Q}^4$ | | $(5.777, 5.225)$ | |
| $\mathcal{Q}^5$ | | $(1.486, 6.120)$ | |
| $\mathcal{R}_{i,k}$ | Square | $\mathbf{y}_{i,k}$ | 0.1 |
| $\mathcal{C}_{i,k}$ | Octagon | $\mathbf{y}_{i,k}$ | 1 |

The optimization problem was solved using the Gurobi® solver with the YALMIP toolbox (Löfberg, 2004) in MATLAB® environment. The computer used had a Intel® i3-8130U (3.4GHz clock) processor and 12GB of RAM. Since we are interested in feasible solutions acquired within reasonable time, the simulations were carried out with three maximum optimization times: 60, 120, and 240 seconds per time step.

### *4.2 Results and Discussion*

The results considering three maximum optimization times: 60, 120, and 240 seconds, are now presented. The corresponding trajectories are depicted in Figures 2, 3, and 4. Each of these Figures is subdivided by approach: a) using constraint (10); b) using constraint (11); c) using the approach presented in Afonso et al. (2020). The evolution of the number of cut vertices over time is presented in Figures 5a, 5b, and 5c.

As shown in Table 2 and Figures 5a, 5b, and 5c, the number of cut vertices of the graph representing the communication network is always zero when either robust connectivity constraint is applied. On the other hand, in the approach of Afonso et al. (2020), the robots spread disregarding the robustness of the communication network, resulting in a fluctuation in the number of cut vertices over time. The trade-off between performance and robustness is evident: the robust approaches perform significantly worse in terms of coverage of targets and overall cost, as show in Figures 2, 3, and 4, and Table 2.

Table 2. Results for each optimization time.

| Approach | Max. solver time (s) | Cost | No. of targets visited | Avg. cut vertices ($\bar{\delta}$) |
|---|---|---|---|---|
| Constraint (10) | 60 | $-1.51$ | 1 | |
| | 120 | $-9.07$ | 2 | 0 |
| | 240 | $-17.34$ | 3 | |
| Constraint (11) | 60 | $-1.49$ | 1 | |
| | 120 | $-17.26$ | 3 | 0 |
| | 240 | $-17.34$ | 3 | |
| Afonso et al. (2020) | 60 | $-20.81$ | 3 | 0.85 |
| | 120 | $-26.42$ | 5 | 1.10 |
| | 240 | $-30.08$ | 5 | 2 |

Regarding the robust approaches, the conservativeness added to the optimization problem due to constraint (11) did not compromise the performance of the resulting trajectories, as compared to (10). In fact, its results were approximately equal or better than the approach using constraint (10), which is less conservative. This superiority is observed in the results with the proposed optimization times, but is unlikely to hold if the periods are substantially increased. However, since closed-loop MPC runs an optimization at each time-step, the performance of an encoding in shorter windows of time is often as important as its asymptotic results. Furthermore, since constraint (11) scales much better than constraint (10), it may be a more adequate option for MAS with a relatively large number of robots.

## 5. CONCLUSION

In this work, mixed-integer programming was employed to plan trajectories for a group of robots. The objective of the group is to navigate through an obstacle-filled environment to collect rewards and reach a terminal target, while maintaining robust connectivity. Two novel constraints were proposed to address the problem of robust connectivity. The results show that, at the expense of performance, the proposed approaches were successful in generating communication networks represented by graphs with no cut vertices. Moreover, for the optimization times considered, approximately equal or better results were observed in the less computationally expensive approach. Future works may address the problems of conservativeness in the robust formulations proposed, as well as improvements in the computational time. Additionally, a possible research line may be the exploration of an algorithm to provide trajectories to the MAS when no feasible solution is found within the maximum optimization time.


## ACKNOWLEDGMENTS

Ângelo Caregnato-Neto acknowledges the support of CAPES for his scholarship. Rubens Afonso acknowledges CAPES (grant 88881.145490-2017-01) and the German Ministry of Education and Research through the Alexander von Humboldt foundation for his fellowship.


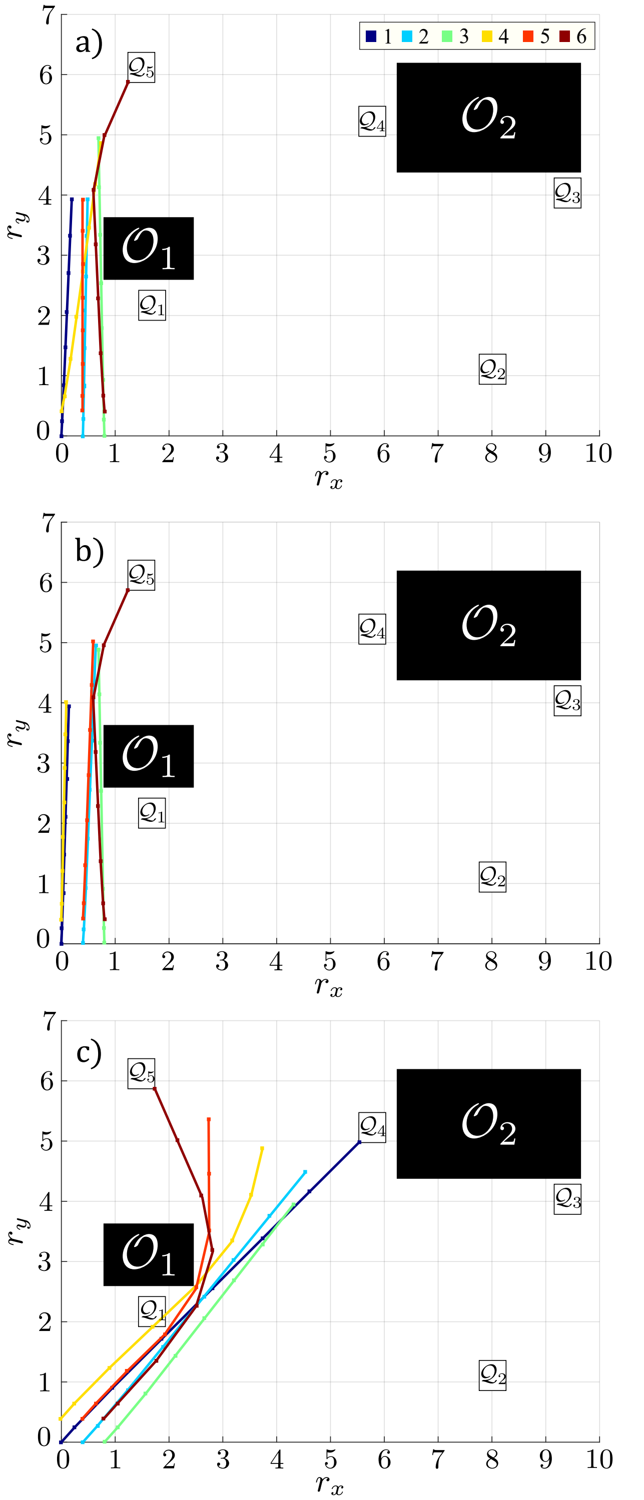


Figure 2. Trajectories with 60s optimization time for a) constraints (10), b) constraint (11), and c) Afonso et al. (2020).

## REFERENCES


Afonso, R.J.M., Maximo, M.R.O.A., and Galvão, R.K.H. (2020). Task allocation and trajectory planning for multiple agents in the presence of obstacle and connectivity constraint with mixed-integer linear programming. *International Journal of Robust and Nonlinear Control*, 30:14, 5464–5491. DOI: 10.1002/RNC.5092.

Agarwal, A., Bhat, S., Gray, A., and Grossmann, I.E. (2010). Automating mathematical program transformation. *Practical aspects of declarative languages (PADL 2010) - Lecture notes in computer science*, 134–148.

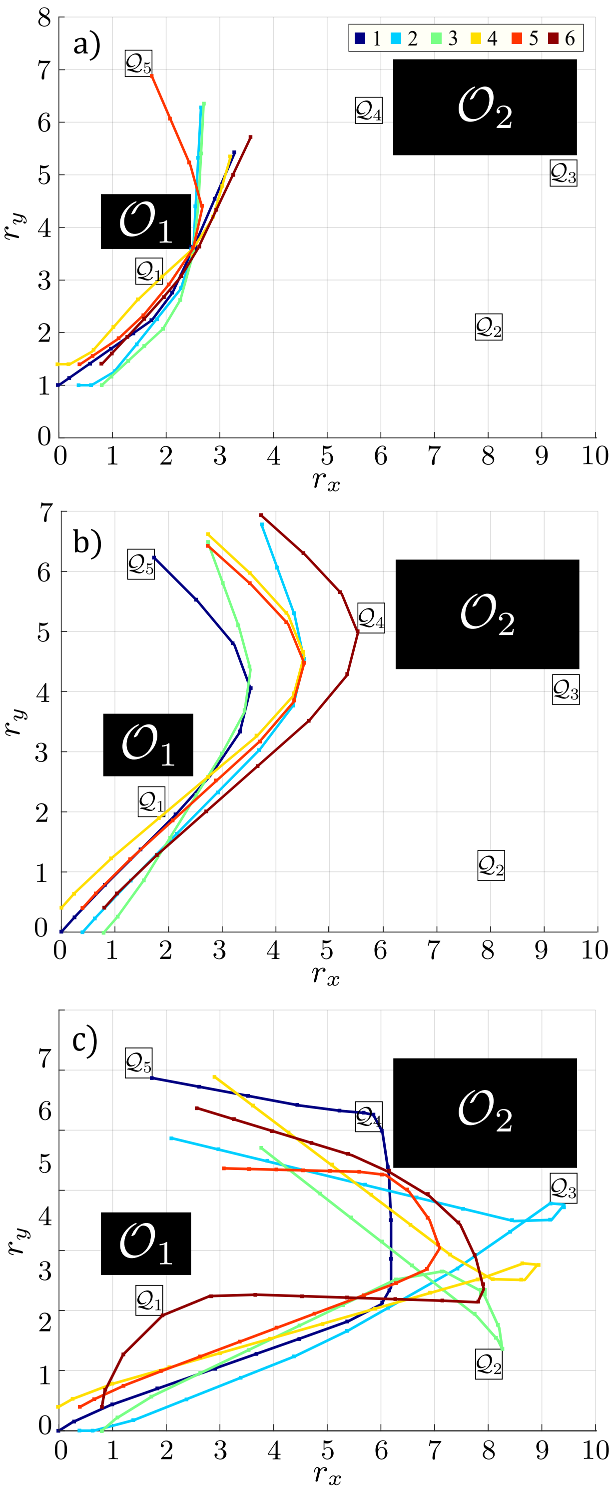


Figure 3. Trajectories with 120s optimization time for a) constraint (10), b) constraint (11), and c) Afonso et al. (2020).

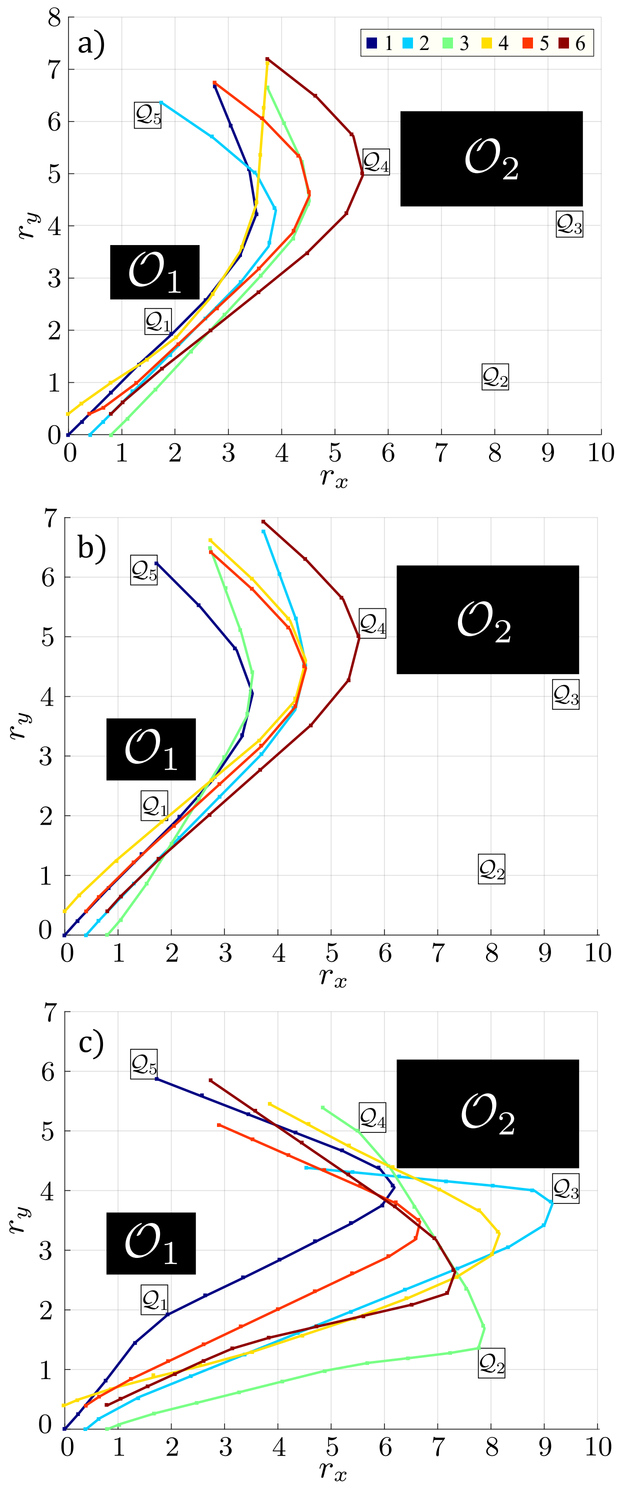


Figure 4. Trajectories with 240s optimization for a) constraints (10), b) constraint (11), and c) Afonso et al. (2020).

Bertrand, G. and Zhang, P. (2012). *A first course in graph theory*. McGraw-Hill Higher Education, Boston.

Dorri, A., Kanhere, S.S., and Jurdak, R. (2018). Multi-agent systems: A survey. *IEEE Access*, 6, 28573–28593.

Filotheou, A., Nikou, A., and Dimarogonas, D.V. (2018). Robust decentralised navigation of multi-agent systems with collision avoidance and connectivity maintenance using model predictive controllers. *International Journal of Control*, 1–15.

Garcia, C., Cardenas, P.F., Saltaren, R., Puglisi, L., and Aracil, R. (2013). Expert systems with applications a cooperative multi-agent robotics system : Design and modelling. *Expert Systems With Applications*, 40:12, 4737–4748.

Harikumar, K., Senthilnath, J., and Sundaram, S. (2018). Multi-uav oxyrrhis marina-inspired search and dynamic

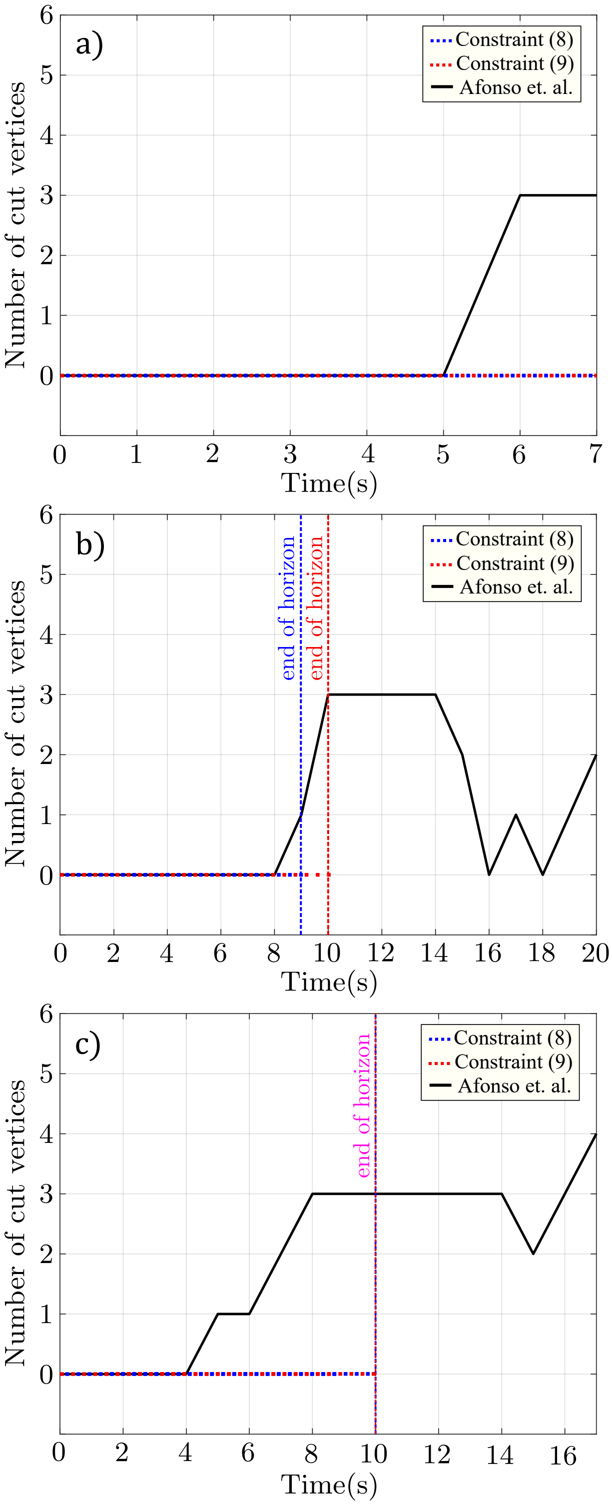


Figure 5. Evolution of number of vertex cuts with a) 60s, b) 120s, and c) 240s optimization time.


formation control for forest firefighting. *IEEE Transactions on Automation Science and Engineering*, 16:2, 863–873.

Klausen, K., Meissen, C., Fossen, T.I., Arcak, M., and Johansen, T.A. (2018). Cooperative control for multirotors transporting an unknown suspended load under environmental disturbances. *IEEE Transactions on Control Systems Technology*, 28:2, 653–660.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*. Taipei, Taiwan.

Ore, O. (1960). Note on hamiltonian circuits. *American math monthly*, 67, 55.

Sun, C., Hu, G., Xie, L., and Egerstedt, M. (2018). Robust finite-time connectivity preserving coordination of second-order multi-agent systems. *Automatica*, 89, 21–27.