BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 69 (73), Numărul 3, 2023
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ
DOI:10.2478/bipie-2023-0015

COMPARATIVE STUDY FOR PATH PLANNING BASED ON THE DECOMPOSITION OF A CO-SAFE LTL SPECIFICATION

BY

IOANA HUSTIU*

“Gheorghe Asachi” Technical University of Iași,
Faculty of Automatic Control and Computer Engineering, Iași, Romania

Received: May 9, 2024
Accepted for publication: August 7, 2024

Abstract. This paper is presenting a comparison between two different methods for decomposing a co-safe LTL global specification that must be accomplished by a team of mobile agents by combining our previous work. Both decomposition algorithms are aiming to break the global mission into independent tasks such that synchronizations between members of the robotic team are not required and the sub-formulas obtained can be accomplished by a single agent. Once the split of the LTL formula is performed, trajectories are projected based on an allocation of tasks to robots, which requires two steps. First time, a cost matrix is computed, followed by the calculation of a binary allocation matrix. The latest will dictate which task, or tasks, will be executed by each mobile robot. The comparison is taking into consideration the variation of the size of the robotic team and the complexity of the global LTL mission, while proposed scenarios and simulations are highlighting the advantages and disadvantages of each decomposition method. The quantitative analysis is involving two criteria: a time related one and a trajectory related one.

Keywords: mobile robots, discrete event systems, path planning, decomposition algorithm, LTL specification.

---

*Corresponding author; *e-mail*: ioana.hustiu@academic.tuiasi.ro

1. Introduction

Autonomous behaviours based on one or more missions that must be accomplished by a team of mobile robots are a highly studied problem in the literature. Paths that are ensuring execution of tasks is commonly approached using high-level specification and one of the regular formalisms is Linear Temporal Logic (LTL) (Kloetzer and Melta, 2008; Fainekos *et al*., 2005; Schillinger *et al*., 2017). The language mentioned is used from decades for verifying properties of discrete event systems (Baier and Katoen, 2008; Clarke *et al*., 1999; Kupferman and Vardi, 2001).

In the stated problem, the mobile agents are evolving in a workspace which can be known, partially known or unknown. There are different approaches in the literature used for modelling the environment and the robotic team, as well as their movement. The concept of transition systems is representing one of the methods for modelling the mentioned notions, but the state explosion problem is likely to happen. In (Ulusoy *et al*., 2012) is presenting an approach towards optimal path planning when using weighted transition systems for a team of robots that must accomplish a mission given as a LTL formula. Another approach to project trajectories for agents is to use probabilistic models (Thompson *et al*., 2009; Ding *et al*., 2014) or to use the concept of Robot Motion Petri Net (RMPN), a model that is scalable with respect to the number of the members of the team (Mahulea *et al*., 2020).

Each technique recount above can consider different strategies regarding the possibility of communication between agents, depending on the necessities and the applications implicated. Recent work in the literature pertaining to this subject is closely related to a compositional technique regarding the reactive synthesis (Finkbeiner *et al*., 2023), while for realistic assumptions when state and communication constraints and limited local view and information are considered (Yu and Dimarogonas, 2021; Chen *et al*., 2024) can be consulted. The authors of the latest paper are generating a plan that is conflict free based on predictions that are using task information from other agents. In this paper the path planning problem is a well-defined goal, the methods discussed are projecting solutions such that synchronizations between agents are avoided since in real-life scenarios the necessity of communications could represent an impediment. In order to achieve such a requirement our approach is to compute trajectories that are unconstrained by other members of the team by dividing the global mission into independent tasks. The focus is on a comparison with respect to the run times on one hand, and to a considered cost on the other hand.

The comparison presented is combining results from (Hustiu *et al*., 2020) and (Hustiu *et al*., 2022). The main contribution is reflected by a detailed comparison between two methods. Despite some limitations in the comparative analysis, this work offers valuable insights into the strengths and weaknesses of each method, providing a foundation for further research and improvement. By highlighting key differences and potential areas for enhancement, this work paves the way for future advancements and more refined comparative studies. The structure of the paper is as it follows: Section 2 is dedicated to theoretical notions that are vital for this work together with a brief description of the decomposition algorithms. Section 3 is completely dedicated to simulations used to discuss and highlight advantages and disadvantages of each method through which the previously mentioned comparison is produced. Finally, conclusions are wrapping up the analysis.

2. Problem overview

This section provides the theoretical background that is supporting the work presented. Sub-section 2.1 is describing the problem framework, while the sub-section 2.2 is introducing to the reader the two algorithms that will be compared in the following section.

2.1. Theoretical context

Let us consider a team of omnidirectional and identical mobile robots $R = \{r_1, r_2, \ldots r_{|R|}\}$ for which their motion is takes place in an environment which is 2*D*, known, static and contains some regions of interest. The set of regions of interest are denoted by $Y = \{y_1, y_2, \ldots y_{|Y|}\}$ and will be exploited in order to define the global mission. The regions are spread into the environment and can be disjoint. The entire workspace is partitioned in a finite number of cells and modelled as a graph afterwards. For the purpose of clarity, we are not considering obstacles in the environment since those cells that would represent static barriers would be simply eliminated from the set through which the mobile agents have access and as a consequence the adjacency matrix would not permit to include the obstacles in the paths of the robots. The set of nodes of this graph is represented by cells, while the arcs are describing common edges between adjacent cells (Habets *et al*., 2006; Mahulea *et al*., 2020; Gonzales *et al*., 2017). In other words, the motion of the mobile agents is tracked, and their final trajectories are projected using the mentioned graph. An example of an environment with 7 regions of interest and 2 mobile agents (the red and blue circles) is depicted in Fig. 1.

The team of mobile robots have the goal of accomplishing a global mission which is given as a Linear Temporal Logic (LTL) formula. The high-level specification is defined over the set of regions of interest and the preference for it is motivated by its expressivity and the easiness of translating missions from colloquial language by considering some temporal operators ($\mathcal{U} - until, \bigcirc - next, \Diamond - eventually,\ \Box - forever$), next to the well-known Boolean ones. In order to satisfy the LTL mission, is necessary to model it as a state machine called

Büchi automaton, a tuple $B = \langle S, S_0, T, \rightarrow_B, S_F \rangle$, where S is the set of states, S_0 is the set of initial states, T is the input alphabet, $\rightarrow_B$ is a non-deterministic transition function and S_F is the set of final states.

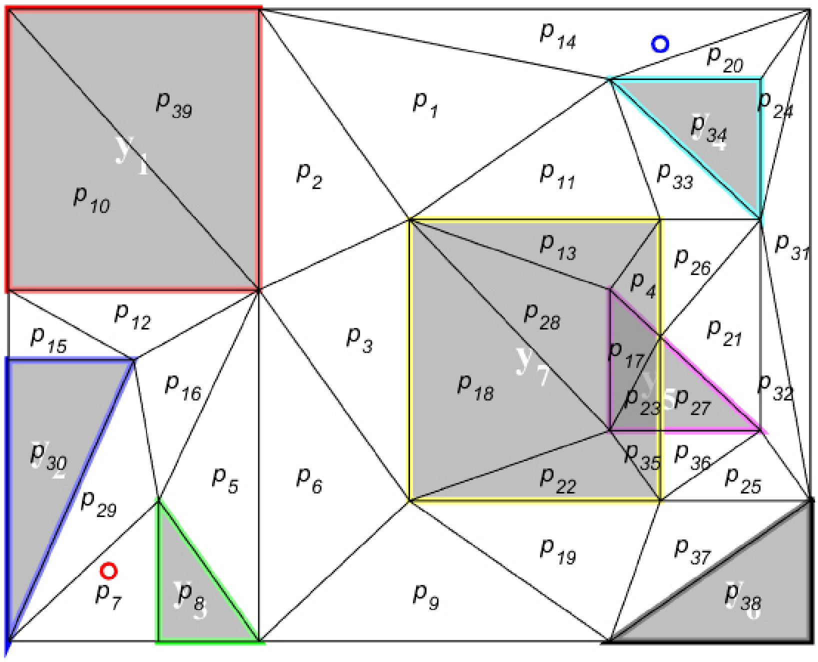


Fig. 1 – Environment with 7 regions of interest and 2 mobile robots.

The Büchi automata accepts or rejects infinite input words that have two components: the prefix, a sequence that conducts from an initial state to an output state, and the suffix, which ensures the satisfiability of the specification. For this paper, we will consider only a sub-class of formulas called co-safe LTL, meaning that from a syntactically point of view, the suffix is irrelevant as it will not affect the value of truth of the formula (Kupferman *et al.*, 2001). In this context and in terms of taking obstacles into account, it can be affirmed that as the blocking cells must be always avoided, the operator $\square - forever$ combined with the Boolean negation $(\neg)$ would provide the desired effect of avoidance (e.g. if we would want to eventually visit regions y_1 and eventually y_2, while avoiding y_3, then a formula such as $\Diamond y_1 \wedge \Diamond y_2 \wedge \square \neg y_3$ should be used). However, the sub-class of co-safe LTL described above is not permitting the usage of such a temporal operator ($\square - forever$) since it would be assumed that a certain condition must be true all the time.

The paper is combining our previous work from (Hustiu *et al.*, 2020) and (Hustiu *et al.*, 2022), while the main contribution in the comparison of the two decomposition algorithms for the global LTL formula. The motivation of comparing is to be able to select one of the methods for extending towards full LTL language as future work. Before describing each algorithm is necessary to explain their role. The main idea of the path planning problem proposed here is to avoid synchronizations between mobile robots by projecting independent trajectories for each of them. Together, the trajectories are ensuring the accomplishment of the global mission and the approach attacked is based on the idea of decomposing the LTL specification in sub-formulas, which will be called tasks, if is possible. Contrary, existing approached that requires communications between agents are available in the literature (Kloetzer and Mahulea, 2015; Kloetzer and Mahulea, 2020).

2.2. Decomposition methods

The two decomposition methods are outlined as will be compared in Section 3. While both algorithms are providing as an output a set of independent tasks, their input is similar. After converting the LTL formula into a Büchi automaton by using the software from (Gastin and Oddoux, 2001), in order to try to break the global task an additional step is necessary: the reduction of the automata. Since it is desired to find sub-formulas that are not requiring more than one agent to be executed, certain transitions will be eliminated, implicitly the ones that cannot be generated by one agent, depending on the environment. This step is also called trimming the Büchi and is crucial for ensuring the lack of synchronizations. As mentioned before, the input for the decomposition algorithms are similar and is represented by the trimmed Büchi automaton. In the following sub-sections, the two methods for dividing the global specification into tasks are briefly described.

2.2.1. Method I

As previously mentioned, the reduced Büchi automata *B* is used as one of the inputs for this method. During algorithm, for each accepted run that conducts from an initial state $s_i \in S_0$ to a final state $s_f \in S_F$, all possible combinations between transitions are sought. If all these permutations are feasible runs of the automaton, then is safe to say that each observation that defines a transition of the initial run will represent an independent task which will be placed in a container. In other words, if elements of a set of observations can be generated in any order and are representing accepted runs of *B*, then a decomposition of the global LTL specification is found. Note that by using the trimmed Büchi automaton, each transition can be enabled by a single agent. A pseudo-code and detailed explanation of the procedure, together with simulation examples are available in (Hustiu *et al.*, 2020)

In addition of not being necessary to produce intermediate synchronizations, the decomposition itself does not depend on the number of members of the robotic team, but there are some limitations that are referring to the sub-class of formulas considered, as explained in (Hustiu *et al*., 2020).

2.2.2. Method II

The second method finds its inspiration in the process mining field, where given a set of activities and a log of these activities, a model of the system that is providing the log is projected according to some relations between activities (Van der Aalst, 2016). Here, the same relations are used in order to establish different connections between tasks. A relevant operation is describing the parallelism $("||")$, is showing which tasks can be executed in parallel and will be used to determine the independence of the tasks. Another important operation is sequencing $("\rightarrow")$ and is describing one of the differences from the method described in the previous sub-section.

In our problem, activities are represented by tasks and the log of activities is constituted by the runs of length two from the Büchi automaton. A log-based relation matrix is computed for all possible observations that can be generated by a single robot and used afterwards to distribute an accepted run of the automata into containers. Each container will include one or more tasks, depending on the sequencing relation. For example, if $task_j$ must be executed after $task_i$ ($task_i \rightarrow task_j$), then those two will be placed into the same container. If a parallelism relation exists between them, then an allocation to different containers will be ensured. The method is described in detail and is accompanied by a pseudo-code and examples in (Hustiu *et al*., 2022).

3. Simulation results

The simulations are made on a computer with $i7-10th$ gen. CPU, $@2.60GHz$ and 16 GB RAM. Before presenting 2 different scenarios used for comparisons is necessary to explain, in essence, the modality of projecting the trajectories, since after decomposing the global mission, we end up with a set of containers that can be independent accomplished, called $TaskSet$. The final paths are obtained after an allocation of containers to robots, procedure that is explained in detail in (Hustiu *et al*., 2021). The main steps are referring at a two-step allocation which requires solving two different Mixed Integer Linear Programming (MILP) problems. A first step is conducting to a cost matrix W

with $|TaskSet|$ rows and $|R|$ columns, which is indicating the cost of each agent to execute each container. The cost is considered to be number of traversed cells that are necessary to accomplish a task. Based on this matrix, a binary allocation matrix $Z \in \{0,1\}^{|R|\times|TaskSet|}$ is projected, where non-zero elements are indicating which container will be fulfilled by each robot (e.g., a non-zero element on row 2 and column 5 means that agent 2 has the job of executing container 5). Note that an agent could end up with more than one task, if necessary.

Scenario 1. Let us consider an environment with 7 regions of interest and 2 mobile agents that must accomplish the global task from Eq. (1).

$$\gamma = \Diamond\, y_4 \wedge \Diamond\, (y_1 \vee \Diamond y_2) \tag{1}$$

The high-level specification which is global, is requiring to visit eventually y_4 and eventually y_2 after reaching region y_1. Regarding its interpretation, it can be observed that a sequencing is necessary. As both decomposition methods are trying to determine sub-formulas that can be generated by a single robot, a dependence of the spread of the regions in the environment appears for the first algorithm presented. A robot that observes simultaneously at least regions y_1 and y_2 is fulfilling the sequencing required. An environment as the one showed in Fig. 2 is permitting the observation described above. Recall that even if the environment has 7 regions of interest defined, only 3 are appearing in the global LTL mission, hence the input alphabet for the automata is composed only from y_1, y_2 and y_7.

The first difference between the solutions found with each method appears at the decomposition step. First algorithm is providing 2 containers in the set of tasks $TaskSet = \{\{y_4\},\{y_1 \wedge y_2\}\}$. The second method is breaking the LTL specification also in 2 containers, $TaskSet = \{\{y_4\},\{y_1 \rightarrow y_2\}\}$. The obtained decomposition is less restrictive since an agent could accomplish the second container by reaching y_1 and y_2 at the same time or in a sequence. As the tasks obtained after decomposition of the global mission are completely independent and are conducting to independent trajectories, the movement of the agents that are executing tasks takes place concurrently. Therefore, the total cost is associated by the maximum number of cells crossed by an agent. For the current scenario, in the case of the first method the cost is equal with 3 as it can be observed in Fig. 2. The red robot is going to cell p_{38} in which is observing $y_1 \wedge y_2 \wedge y_7$, observation that is including the content of the second container. In Fig. 3, the trajectories projected for agents by using the second decomposition method are ensuring a cost equal with 2. In terms of cost, the difference is not remarkable since it is desired to find a solution instead of not finding one as in real life scenarios it could be more important to have the possibility to exploit different opportunities. The solutions obtained for the described scenario are not presenting significant differences, but a more detailed analysis will be performed in the following.

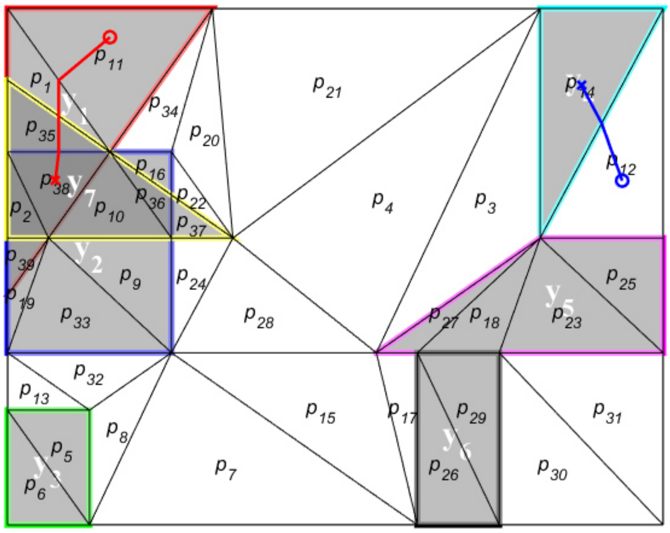


Fig. 2 – Trajectories obtained with method 1 that are fulfilling the global task $\gamma = \lozenge\, y_4 \wedge \lozenge(y_1 \vee \lozenge y_2)$.

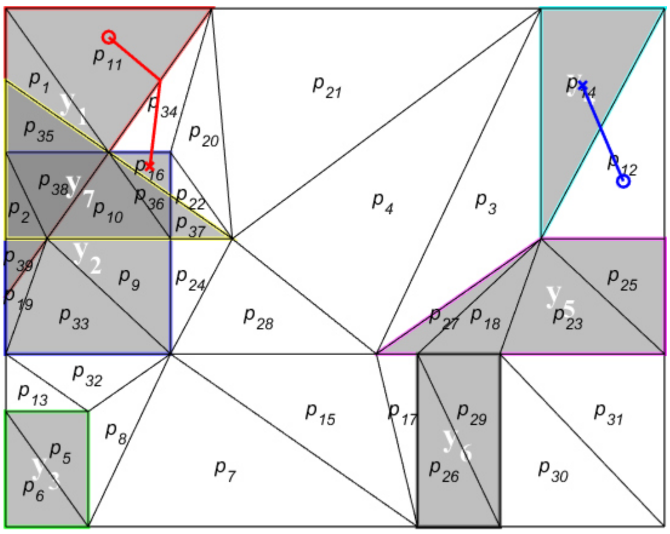


Fig. 3 – Trajectories obtained with method 2 that are fulfilling the global task $\gamma = \lozenge\, y_4 \wedge \lozenge(y_1 \vee \lozenge y_2)$.

Scenario 2. Let us consider an environment with 7 regions of interest and 2 mobile robots that are evolving in it and have the goal of accomplishing the global task from Eq. (2)

$$\gamma = \Diamond y_1 \wedge \Diamond (y_2 \vee y_3) \wedge \Diamond y_4 \wedge \Diamond (y_5 \wedge y_7) \wedge \Diamond y_6 \tag{2}$$

The meaning of the LTL formula is to visit eventually y_1, eventually y_2 or y_3, eventually y_4, eventually the common area of y_5 and y_7 and eventually y_6. In this case, intuitively, we are expecting to obtain 5 independent tasks that will be allocated to the robotic team. Since no sequencing is required, both decomposition algorithms will provide the same output, meaning that the variable $TaksSet$ will have exactly 5 independent containers, namely $TaskSet = \{\{y_1\}, \{y_2 \vee y_3\}, \{y_4\}, \{y_5 \wedge y_7\}, \{y_6\}\}$. Fig. 4 depicts the final trajectories for the agents as the cost matrix is equal with $W = \begin{pmatrix} 4 & 3 \\ 1 & 6 \\ 9 & 2 \\ 7 & 5 \\ 7 & 7 \end{pmatrix}$ and the allocation matrix is $Z = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$. The interpretation of the last matrix is that the first robot (red circle) will accomplish containers 2 and 5 by visiting regions $y_2 \vee y_3$ and y_6 and the second robot (blue circle) will satisfy containers 1, 3 and 4. The order of visiting the regions is dependent on the cost of the tasks as their execution will be in an ascending succession (e.g. if multiple containers are assigned to the same robot, the one with a smaller cost will be accomplished first etc.).

For the current simulation, the run time for obtaining and trimming the Büchi automaton is approximatively equal with 2.38 seconds. The first difference when comparing both methods appear when decomposing the global LTL specification. For the first method, the decomposition took about 5 second, while the run time for the second method was only 0.02 seconds. The discrepancy is related to the fact that first method is analysing all accepted run of the Büchi automata, which has 32 states and conducts to a number of accepted input words equal with 120.

The quantitative comparation of the solutions will be formulated from two different criteria:

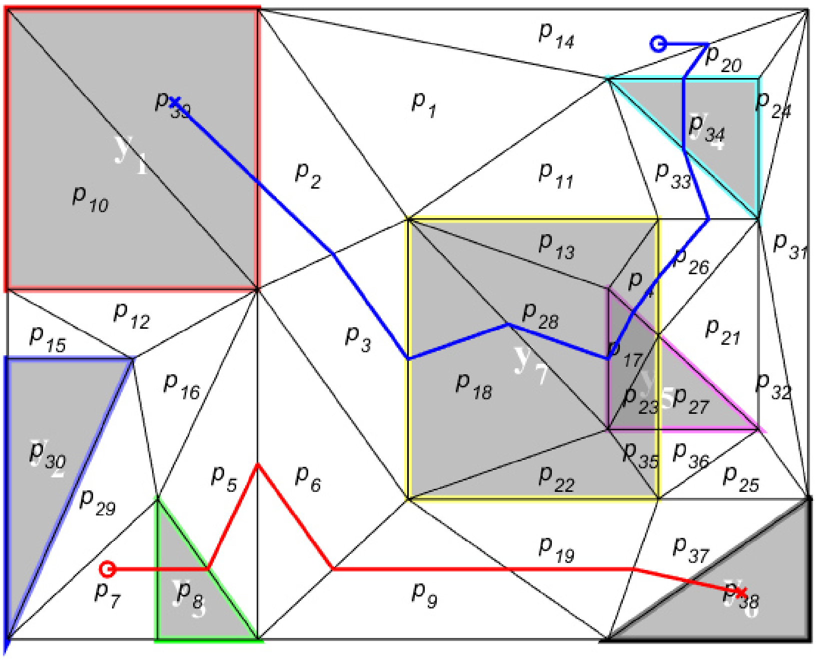
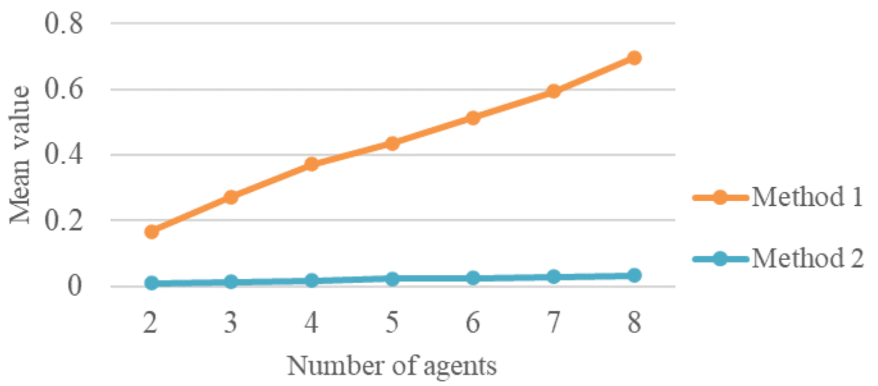


Fig. 4 – Trajectories for a team of two mobile robots that are satisfying the global task γ .

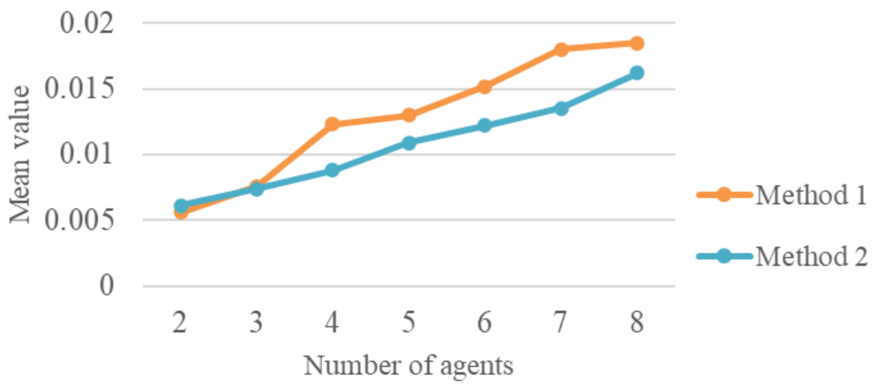
- time related: by analysing the run times to compute the cost matrix, the allocation matrix and the final trajectories.
- trajectory related: by comparing the cost of fulfilling the global LTL task.

In order to provide a meaningful comparison, 100 experiments were conducted for each variation of the size of the robotic team from 2 to 8. The initial position for the agents is randomly chosen and the charts from Fig. 5 (a), (b) and (c) are in a direct correlation to the first item analysed by presenting the mean times for computing the cost matrix, the allocation matrix and for projecting the final trajectories. It can be observed that significant differences can be spotted only for the computation of the cost matrix. As we are looking ar Fig. 5 (d), where the mean cost is graphically illustrated, we can also affirm that there is no gain in the usage of the first decomposition method or the second one.

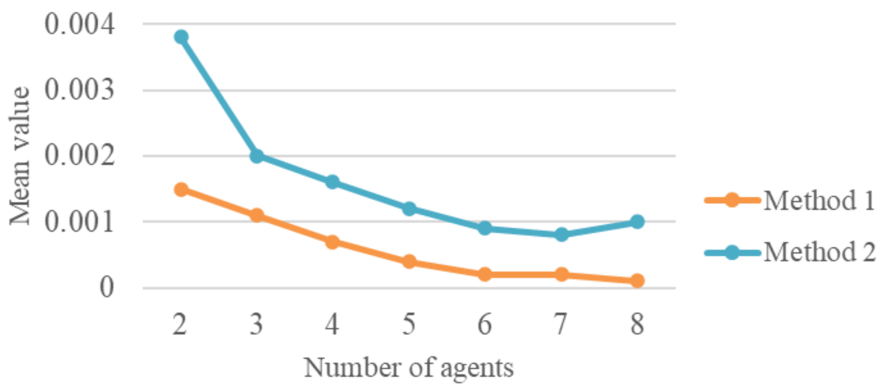
Even if the simulations are pointing out the superiority of the second method in terms of expressivity in the first place, and the run times afterwards, both methods are appropriate when referring to scalability. Given the fact that the projection of the trajectories is based on an offline framework and in such a context, a gain of approximatively 1 second is not necessarily representing a real gain. The results presented above are also confirmed by other simulations for which different complexity of the LTL specification is taken into consideration, together with a variation of the size of the robotic team.



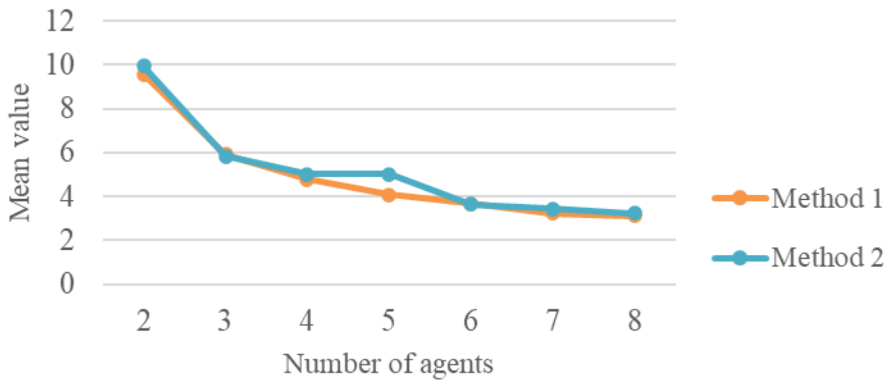
(a) – Cost matrix



(b) – Allocation matrix



(c) – Final trajectories



(d) – Cost

Fig. 5 – Mean values of computation time and cost.

4. Conclusions

The paper is presenting a quantitative comparison between two methods used to divide a global mission into independent tasks such that independent trajectories for each member of the robotic team are projected. The mission is actually a high-level specification in LTL language and must be fulfilled by the team as a whole. For analysing differences between the two algorithms, two scenarios are proposed with a time related and a cost related evaluation performed. After obtaining the set of independent tasks, its content is allocated to agents by solving two MILP problems. The trajectories computed are ensuring the avoidance of synchronizations between robots which is a strong advantage in the context of a real-life scenario.

Future work will consider the extension towards full LTL language by finding mechanisms to handle the necessity of multiple sequencing. Also, for conducting real-life experiments the collision avoidance problem will be taken into account.

Acknowledgements. I would like to thank my supervisors, Professor Marius Kloetzer and Professor Cristian Mahulea for their invaluable contributions to the development of the methods compared in this study. Their expertise and insights were instrumental in guiding my research and ensuring the rigor of the comparative analysis.

REFERENCES

Baier C., Katoen J.P., *Principles of model checking*, MIT Press, 2008.

Chen Z., Li L., Kan Z., *Distributed Task Allocation and Planning under Temporal Logic and Communication Constraints*, IEEE Robotics and Automation Letters, 9(7), 2024.

Clarke E.M.M., Peled D., Grumberg O., *Model checking*, MIT Press, 1999.

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

Fainekos G.E., Kress-Gazit H., Pappas G.J., *Hybrid controllers for path planning: A temporal logic approach*, Proc. of the 44th IEEE Conference on Decision and Control (CDC), IEEE, 4885-4890, (2005).

Finkbeiner B., Geier G., Passing N., *Specification decomposition for reactive synthesis*, Innovations in Systems and Software Engineering, 19(4), 339-357, (2023).

Gastin P. Oddoux D., *Fast LTL to Büchi automata translation*, Proc. of Computer Aided Verification: 13th International Conference (CAV), 2001, Springer Berlin Heidelberg, 53-65.

Gonzalez R., Kloetzer M., Mahulea C., *Comparative study of trajectories resulted from cell decomposition path planning approaches*, Proc. of the 2017 21st International Conference on System Theory, Control and Computing (ICSTCC), 2017, IEEE, 49-54.

Habets L.C.G.J.M., Collins P.J., van Schuppen J.H., *Reachability and control synthesis for piecewiseaffine hybrid systems on simplices*, IEEE Transactions on Automatic Control, 51(6), 938-948, (2006).

Hustiu S., Hustiu I., Kloetzer M., Mahulea C., *LTL task decomposition for 3D high-level path planning*, Journal of Control Engineering and Applied Informatics, 23(3), 76-87, (2021).

Hustiu I., Kloetzer M., Mahulea C., *Distributed path planning of mobile robots with LTL specifications*, Proc. of the 24th International Conference on System Theory, Control and Computing (ICSTCC), 2020, IEEE, 60-65.

Hustiu I., Mahulea C., Kloetzer M., *Distributing Co-safe LTL Specifications to Mobile Robots*, Proc. of. the 26th International Conference on System Theory, Control and Computing (ICSTCC), 2022, IEEE, 306-311.

Kloetzer M., Belta C., *A fully automated framework for control of linear systems from temporal logic specifications*, IEEE Transactions on Automatic Control 53(1), 287-297, (2008).

Kloetzer M., Mahulea C., *LTL-based planning in environments with probabilistic observations*, IEEE Transactions on Automation Science and Engineering, 12(4), 1407-1420, (2015).

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

Kupferman O., Vardi M.Y., *Model checking of safety properties*, Formal Methods in System Design, 19(3), 291-314, (2001).

Mahulea C., Kloetzer M., González R., *Path planning of cooperative mobile robots using discrete event models*, John Wiley & Sons, 2020.

Schillinger P., Bürger M., Dimarogonas D.V., *Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems*, The International Journal of Robotics Research, 37(7), 818-838, (2017).

Thompson S., Horiuchi T., Kagami D., *A probabilistic model of human motion and navigation intent for mobile robot path planning*, Proc. in 4th International Conference on Autonomous Robots and Agents, 2009, IEEE, 663-668.

Ulusoy A., Smith S.L., Ding X.C., Belta C., *Robust multi-robot optimal path planning with temporal logic constraints*, International Conference on Robotics and Automation, 2012, IEEE, 4693-4698.

Van Der Aalst W., *Process mining: data science in action. Vol. 2.*, Heidelberg: Springer, 2016.

Yu P., Dimarogonas D.V., *Distributed motion coordination for multirobot systems under LTL specifications*, IEEE Transactions on Robotics, 38(2), 1047-1062, (2021).

STUDIU COMPARATIV PENTRU PROIECTAREA TRAIECTORIILOR OBȚINUTE PRIN DESCOMPUNEREA UNEI SPECIFICAȚII LTL

(Rezumat)

Această lucrarea prezintă o comparație între două metode diferite de a descompune o specificație globală care face parte din clasa logicii temporal lineare (LTL) de tip co-safe, care trebuie îndeplinită de o echipă de agenți mobili, luând în considerare rezultatele noastre anterioare. Cele două metode au ca scop descompunerea acestei misiuni globale în taskuri independente astfel încât între membrii echipei robotice să nu fie necesare sincronizări, iar fiecare dintre taskuri pot fi executate de un singur agent. Odată ce sub-formule independente ale misiunii inițiale sunt determinate, traiectoriile pentru agenți sunt proiectate pe baza unei alocări a taskurilor către roboți, care presupune doi pași. Primul dintre ei este reprezentat de calculul unei matrice de cost, iar cel de-al doilea este descris de o matrice binară de alocare. Comparația prezentată ia în considerare variații ale numărului de membri ai echipei robotice, în timp ce scenariile și simulările evidențiază avantajele și dezavantajele fiecărei metode de descompunere.