Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 9(5): 11511–11536.
DOI: 10.3934/math.2024565
Received: 03 December 2023
Revised: 16 January 2024
Accepted: 19 March 2024
Published: 25 March 2024

Research article

On the use of fuzzy preorders and asymmetric distances for multi-robot communication

Jose Guerrero[1,2], **Javier Antich**[1,2,*] **and Oscar Valero**[1,2]

[1] Mathematics and Computer Science Department, Universitat de les Illes Balears, Carr. Valldemossa km. 7.5, Palma, 07122, Spain

[2] Health Research Institute of the Balearic Islands (IdISBa), Hospital Universitari Son Espases, Palma, 07120, Spain

* **Correspondence:** Email: javier.antich@uib.es.

Abstract: One of the main problems to be addressed in a multi-robot system is the selection of the best robot, or group of them, to carry out a specific task. Among the large number of solutions provided to allocate tasks to a group of robots, this work focuses on swarm-like approaches, and more specifically on response-threshold algorithms, where each robot selects the next task to perform by following a Markov process. To the best of our knowledge, the current response-threshold algorithms do not provide any formal method to generate new transition functions between tasks. Thus, this paper provides, for the first time, a mathematical model, as based on the so-called fuzzy preorders, for the allocation of tasks to a collective of robots with communication capabilities. In our previous work, we proved that transitions in the aforementioned process can be modeled as fuzzy preorders, constructed through the aggregation of asymmetric distances, in such a way that each robot makes its decision without taking into account the decisions of its teammates. Now, we extend this model in such a way that each robot will take into account the number of robots previously allocated for each task. To implement this method, a very simple communication mechanism has been considered. Several simulations have been carried out in order to validate our approach. The results confirm that fuzzy preorders are able to model the evolution of the system when this type of communication is considered and show when and how the communication process improves the system's performance. Experimental results show the existence of a set of good values for the maximum communication distance between robots and that these values depend on the distribution of the tasks in the environment. Thus, in some cases, a better communication mechanism does not imply better results.

Keywords: multi-robot communication; fuzzy preorders; asymmetric distances; task allocation; possibility theory; aggregation
Mathematics Subject Classification: 54E35, 54H30, 68T37, 93A16

1. Introduction

Multi-robot systems, defined as a group of two or more robots with a common mission, provide a great number of advantages over systems with only one robot. In contrast, several open problems must be addressed to fully exploit those advantages. Among these problems, this paper will be focused on task allocation issues, that is, how each robot decides the next task to execute. Swarm intelligence methods, which are inspired by the behavior of colonies of insects, give us a very appropriate approach to solve the aforementioned problems with very low computational constraints. In such systems, behavior resembling global intelligence emerges from the interaction of very simple behaviors that are executed by every robot. Nowadays, there are several approaches that are inspired by swarm-like paradigms, among the most widely used approaches are the so-called response-threshold methods (RTMs) [1,2]. The RTMs assign a value, called a stimulus, to each robot-task pair in order to model the adequacy level of the robot to suit the task. For example, the stimulus can be the inverse of the distance between the task and the robot. Each robot decides the next task to perform according to a probability that depends on the aforementioned stimulus and the task in which the robot is currently located. Thus, the aforementioned probabilities are modeled by using the so-called probabilistic transition functions and, at the same time, the system can be modeled as a memoryless process, i.e., a probabilistic Markov chain, where the states are the tasks and the transition probabilities (modeled by means of the transitions functions) depend on the stimulus value.

In general, the communication mechanism between robots plays a key role in task allocation. Some allocation algorithms, like for example those derived from auctions [3], are based on explicit communication protocols between robots to implement negotiation processes. Auction methods, and negotiation paradigms in general, provide better solutions than swarm approaches. Nevertheless, the communication requirement in such methods can become a drawback. In order to overcome this problem, some papers, such as [4–6], have focused on the behavior of those methods under communication restrictions. In any case, the proposed solutions still require complex communications protocols. In contrast, swarm-like mechanisms do not require explicit communication protocols, but the use of some communication can improve the system's performance, as was analyzed in [7]. Some works in which specific examples of swarm algorithms that use very simple communication mechanisms are as follows: Deshpande et al. [8] proposed a swarm algorithm whereby each robot broadcasts its position to coordinate multiple robots to form specific geometric patterns (as circles) around the location of the targets; in [9] the robots communicate their internal state through colored LEDs; finally, in [10] the robots also communicate their positions in order to allocate the targets; they also follow a given leader. In [11] Talamali et al. make use of a communication mechanism to implement swarm-like task allocation methods. Here, each robot combines, via a voting process, the information obtained from its local sensors with the information received from its neighbors to reach a collective decision. Despite the authors presenting a schematic formal analysis corresponding to when this consensus is reached, this method is more complex than the RTM approach implemented in our paper. Moreover, how to aggregate the data coming from different sources is not analyzed in [11]. Finally, in [12] Gielis et al. conducted an exhaustive review of communications in multi-robot systems, but without providing any formal method to aggregate the communication to the decision process. In the light of the exposed facts, one of the goals of this paper is to propose a formalism to generate transition functions that take into account a very simple communication between robots. As will be seen later, in the case in which the communication between robots is taken into account, the transition function of an RTM algorithm can be modeled as a fuzzy preorder.

In [13], we introduced a new theoretical formalism for RTMs based on possibility theory in the sense of [14, 15]. Thus we developed a new RTM which incorporates possibility theory instead of probability for task allocation from a very general agent perspective without taking into account the specific characteristics of the robots. The evolution of the system was modeled as a possibilitic Markov chain in which transition probabilities were replaced by transition possibilities and, in addition, the classical sum-product algebra was replaced by the max-min one (for a fuller treatment of the topic we refer the reader to [16]). Inspired by this fact, in [1], all transition functions (transition probabilities) considered in the literature were identified by using transition possibilities through the use of indistinguishability operators (see [17] for a detailed treatment of indistinguishability operators). In this way, a formal methodology for generating transition possibilities via indistinguishability operators was developed; thus, new and different transition functions based on indistinguishability operators were introduced and shown to be appropriate mathematical tools in order to model possibilistic response functions and describe the evolution of multi-robot systems according to different situations addressed in [18]. This was, to the best of our knowledge, the first formal and systematical method to generate response functions for multi-robot systems in the literature. In all cases analyzed in the preceding references, the possibilistic approach was compared to the probabilistic one in such a way that the formal and empirical results evidenced that the former always outperforms the latter; particularly, it converges faster to a stationary state, it requires lower computational capacities, and so on. It must be pointed out that, in the aforementioned paper, thanks to indistinguishability operators, a formal method to generate possibilistic response functions from distances and utilities was stated by using the aggregation of asymmetric distances in the sense of [19].

In this paper, we extend our previous work developed in [18]. Thus we introduce a mathematical model based on fuzzy preorders i.e., a generalization of indistinguishability operators, with the aim of taking into account the impact of the communication between the agents of the multi-robot system on robots' behavior. Hence, each robot broadcasts its location to its teammates located within a neighborhood. This information is aggregated to the distance information and incorporated in the transition possibility function. Thus, each robot will know how many robots are allocated near each task and, according to this information, it will perceive based on the stimulus the adequacy level of the tasks. Hence tasks with a high number of allocated robots or are very distant will have a very low transition possibility function values. In contrast, the smaller the number of robots or the closer the tasks, the higher the values of the transition possibility functions. As will be proved later in this paper, the new transition possibility functions are exactly fuzzy preorders and, thus, all of the aforementioned theoretical background can be applied to them. To the best of our knowledge, this is the first study that provides a formal method to create new transition functions that take into account the interaction (communication) between robots.

The task performed to validate our approach is defined as follows: The robots should visit a group of tasks placed in the environment. Initially, each robot is allocated to a task and it must decide the next one to execute by following a possibilistic transition function. It must be stressed that the values of the transition functions change over the execution time (and hence the stimulus associated with each task and perceived by the robot) and the tasks are not removed. Thus, a task can be visited by several robots at many time instants. The goal of the system is to reduce the number of steps required to visit, at least once, each task. The large number of experiments, performed by using Matlab, show that the use of a limited amount of communication improves the system results under certain circumstances. Moreover, we have studied the impact of the task distribution on the system's performance. The experimental results show that there is a correlation between the system's performance (the aforementioned required number of steps to visit the tasks) and the maximum communication distance. In some cases, as shown and discussed in Section 4, a very large communication range degrades the system's performance. However, an appropriate communication distance, which improves the system's performance, always exists for each placement or distribution of the tasks in the environment.

Thus, the above-mentioned contributions of this paper can be summarized as follows:

- This paper introduces a possibilistic/fuzzy framework for modeling the communication mechanism among robots.
- For the first time, we propose a formalism based on indistinguishability (fuzzy preorders) to aggregate information obtained from other robots.
- The experiments detailed in this paper demonstrate the benefits of robot communication. However, depending on the specific environment characteristics, excessive communication can adversely affect system's performance.
- Furthermore, when communication enhances system performance, our method exhibits greater robustness in terms of the parameters.
- In the future, we will present methods for generating new fuzzy preorders that allow one to describe the behavior of multi-robot systems, adjusting appropriately according to the characteristics of both the environment and tasks.

The structure of the remainder of the paper is as follows: In Section 2, the main concepts on RTMs based on fuzzy preorders and aggregation of the asymmetric distances and possibilites are introduced, as well as the key previous work performed by the authors of [18]; Section 3 explains the communication process between robots and how the fuzzy preorders used in [18] must be modified in order to include the new information provided by the aforementioned process; in Section 4, we will show the experimental results. Finally, Section 5 presents the conclusions and future work.

2. Previous work: Aggregation of asymmetric distances and possibilistic approach to task allocation

As was aforementioned, nowadays one of the most widely used swarm-like methods are those based on the so-called RTMs, where there are a set of $nr \in \mathbb{N}$ robots $R = \{r_1, ..., r_{nr}\}$ and a set of $nt \in \mathbb{N}$ tasks to be visited $T = \{t_1 ..., t_{nt}\}$. Notice that $\mathbb{N}$ stands for the set of positive integer numbers. In these methods, each robot $r_k \in R$ has associated with each task t_j to be executed a stimulus ($s_{r_k,t_j} \in \mathbb{R}$), where $\mathbb{R}$ denotes the set of real numbers. Those stimuli represent the suitability of the mentioned task for the robot. For example, the stimulus can be the inverse of the distance between the task and the robot. According to [20, 21], when the robot is located at the task t_i and s_{r_k,t_j} exceeds a threshold value θ_{r_k} ($\theta_{r_k} \in \mathbb{R}$), the robot r_k starts the execution of t_j by following the probability $p(r_k, ij)$ given as follows:

$$p(r_k, ij) = \frac{s^n_{r_k,t_j}}{s^n_{r_k,t_j} + \theta^n_{r_k}}, \tag{2.1}$$

where $n \in \mathbb{N}$.

Notice that in the previous expression we are assuming that robot r_k is allocated in the task t_i and the possibility to leave task t_i in order to execute task t_j is being evaluated. Moreover, we are assuming that the threshold θ_{r_k} depends only on the robot r_k and that, hence, for each robot, such a value is the same for all tasks.

In [18], fuzzy preorders, in the sense of [17,22], were shown to be an appropriate mathematical tool to model response functions satisfying the expression given by (2.1).

Let us recall, according to [22], that a fuzzy preorder is a fuzzy set, in the sense of [14] with $E : X \times X \to [0, 1]$ satisfying the following conditions for each $x, y, z \in X$:

(i) $E(x, x) = 1$; (Reflexivity)

(ii) $E(x, z) \geq T(E(x, y), E(y, z))$. (Transitivity)

It must be stressed that in the transitivity axiom, T is a triangular norm (t-norm for short) and that for the basics of t-norms we refer the reader to [23].

Moreover a fuzzy preorder E is said to separate points provided that it satisfies, for each $x, y \in X$, the following condition **(i')**:

(i') $E(x, y) = E(y, x) = 1 \Rightarrow x = y$.

In [18], the considered stimulus was a constant value depending only on the (Euclidean) distance among tasks and a fixed utility associated with each task. The utility of a task t_j was defined as a numerical value $U_j \in \mathbb{R}^+$ ($\mathbb{R}^+$ denotes the set of non-negative real numbers) that represents the importance of performing that task. The higher the utility value, the more important the task. The resulting (possibilistic) transitions between tasks are given as follows:

$$p(r_k, ij) = \frac{\theta_{r_k}^n}{\theta_{r_k}^n + Q_{\Phi,q,d_E}^n((U_i, x_i, y_i), (U_j, x_j, y_j))}, \tag{2.2}$$

where θ_{r_k} is the threshold constant for each robot r_k. Observe that, following [18, 19], $Q_{\Phi,q,d_E} : \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}^+$ is an asymmetric distance that depends on the utility of each task (U_i and U_j) and the Euclidean distance $d_E((x_i, y_i), (x_j, y_j))$, where (x_i, y_i) and (x_j, y_j) denotes the allocation coordinates of tasks t_i and t_j, respectively. Notice that $\mathbb{R}^2 = \{(x, y) : x, y \in \mathbb{R}\}$.

It must be pointed out that an asymmetric distance satisfies all metric axioms except for the symmetry (see, for instance, [24]). Concretely, following [24], an asymmetric distance on a (non-empty) set X is a function $d : X \times X \to [0, \infty]$ such that the following holds for all $x, y, z \in X$:

(AD1) $d(x, x) = 0$,

(AD2) $d(x, z) \leq d(x, y) + d(y, z)$.

Clearly every metric is an asymmetric distance but the converse is not true. In the light of the exposed notion, it must be pointed out that the asymmetric distance Q_{Φ,q,d_E} is defined, for all (U_i, x_i, y_i) and (U_j, x_j, y_j), by

$$Q_{\Phi,q,d_E}((U_i, x_i, y_i), (U_j, x_j, y_j)) = \alpha_u \cdot q_U(U_i, U_j) + d_E((x_i, y_i), (x_j, y_j)), \tag{2.3}$$

where $q_U(U_i, U_j) = \max\{U_j - U_i, 0\}$ matches up with an asymmetric distance which measures the improvement in utility made when a task t_i is abandoned by the robot and it starts to perform the task t_j. Hence the value $\max\{U_j - U_i, 0\} = 0$ can be interpreted because the task t_i is more attractive than the task t_j. Notice that the improvement in utility cannot be provided a metric. Observe that the distance Q_{Φ,q,d_E} is constructed by means of an aggregation of the asymmetric distance q_U and a metric d_E.

It is worth mentioning that α_u is a system parameter that, on the one hand, causes the utility value to have the same dimension and scale as the distance and, on the other hand, indicates the importance of the utility with respect to the distance. We refer the reader to [18] for a deeper discussion of the role played by α_u.

In light of the exposed fact, it seems natural to expose the reason for which Q_{Φ,q,d_E} is exactly an asymmetric distance. The answer to the posed question is given by the following result (see [19, Theorem 6]).

Proposition 1. *Let $\Phi : (\mathbb{R}^+)^2 \to \mathbb{R}^+$ and let d_1, d_2 be two asymmetric distances defined on non-empty sets X and Y, respectively. Then the following assertions are equivalent:*

1) *The function $Q_{\Phi,d_1,d_2}: (X \times Y)^2 \to \mathbb{R}^+$ is an asymmetric distance, where $Q_{\Phi,d_1,d_2}(x, y) = \Phi(d_1(x_1, y_1), d_2(x_2, y_2))$ for all $x = (x_1, x_2), y = (y_1, y_2) \in X \times Y$.*

2) *The function Φ is monotone, subadditive and $\Phi(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2 = 0$.*

Clearly, the function $\Phi : (\mathbb{R}^+)^2 \to \mathbb{R}^+$ given, for all $x, y \in (\mathbb{R}^+)^2$, by $\Phi(x, y) = \alpha_u \cdot x + y$ satisfies assertion 2 in the statement of Proposition 1; thus, the function Q_{Φ,q,d_E} is exactly an asymmetric distance on $\mathbb{R}^+ \times \mathbb{R}^2$.

Finally, it must be stressed that, on account of Proposition 1 in [18], given $\theta \in \mathbb{R}^+$ with $\theta > 0$, $n \in \mathbb{N}$ and an asymmetric distance q on a non-empty set X, then the fuzzy set $E^n_{\theta,q,Dom} : X \times X \to [0, 1]$, defined by

$$E^n_{\theta,q,Dom}(x, y) = \frac{\theta}{\theta + (q(x, y))^n} \tag{2.4}$$

is a fuzzy preorder that separates points when the t-norm under consideration, for the transitivity axiom, is the Dombi t-norm $T^{\frac{1}{n}}_{Dom}$, where

$$T^{\frac{1}{n}}_{Dom}(a, b) = \begin{cases} 0, & \text{if } a = 0 \text{ or } b = 0 \\ \frac{1}{1+\left(\left(\frac{1-a}{a}\right)^{\frac{1}{n}}+(\left(\frac{1-b}{b}\right)^{\frac{1}{n}}\right)^n}, & \text{elsewhere} \end{cases} . \tag{2.5}$$

In light of the preceding remark we can deduce immediately that the transition probability given by expression (2.2) matches up with a fuzzy preorder that separates points and, thus, that those transition values given by $p(r_k, ij)$ can be understood as fuzzy preorder and, hence, as possibilities.

Possibilistic approaches provide a large number of advantages over the classical probabilistic framework. Some of them, already outlined in Section 1, are as follows:

- In general, the transition functions $p(r_k, ij)$ provided by (2.2) from one state (task) t_i to the other ones t_j do not satisfy the conditions of the axioms of a probabilistic distribution. Concretely, the sum of all the values is not equal to 1, i.e., $\sum_j p(r_k, ij) \neq 1$. The probabilistic approaches must perform unnatural manipulations in order to transform the set of transitions values to a probabilistic distribution. For example, each value must be normalized and from the new resulting distribution each robot will select the next task to execute. Thus, the characteristics of the Markov chain resulting from the normalization have changed from those of the original process. In the case that the values of the transition functions are modeled as a fuzzy set, and, therefore, the case that the system is modeled as a fuzzy Markov chain, there are several methods that have been developed to make a decision without changing the original values (see [25]).

- To the best of knowledge, there does not exist any previous method to generate transition functions. Previous works in the literature (see, for example, [26]) sugest the use of ad-hoc functions for a specific task or environment, without following any methodical process. Thanks to using model transition functions as fuzzy preorders, as it is proposed in our paper, the system designer has a theoretical background to systematically generate new transition functions which could be better adapted to the designer's proposal. According to the exposed theoretical background, a few new possibilistic transition functions have been introduced, and their utility in multi-agent systems has also explored, in [27, 28].

3. Task allocation and communication between agents

This section is devoted to explaining the task allocation process when a simple communication mechanism is taken into account. The main objective is to analyze how the aggregation of fuzzy preorders and asymmetric distances allows us to model the evolution of the system when communication is under consideration and, in addition, to show when and how the communication process improves the system's performance. To this end, as was exposed in Section 1, this work modifies the transition function given by (2.2) to allow the communication between robots (how such a modification is made will be subsequently explained in detail). From now on, in the simulations and for all forthcoming processes and algorithms, we have defined a time unit, which we have called a step, in such a way that, in each simulation step, the robot makes the decision (i.e., it decides the next task to execute). In order to perform the task allocation, each robot must select the task and, in addition, it must communicate the selected task to all other robots. The evolution of the system is due to the execution of two processes: The task processing process and the communication processing process. In the following subsection, we explain both processes.

3.1. Task execution and communication

Figure 1 summarizes the steps to be followed by a robot. On the one hand, the left side of this figure depicts the steps executed by the robot to carry out its current task and select the next one. On the other hand, the right side of the figure specifies how the robot receives and processes the messages from other robots. Both sides, task processing and communication processing, are performed in parallel.

During the task processing, the robot executes the following sequential steps after arriving at a task:

(1) *Task processing*: After getting to the task position (task t_i), the robot starts its processing. As the goal of the study was to analyze the communication issues, this step is assumed to take only one time unit or step. It must be noted that this task does not disappear after being processed by a robot, but it remains in the environment. This is due to the fact that several robots may need to execute the same task.

(2) *Selection of a new task*: In this step, the robot chooses the next task to be executed by following the memoryless process based on the transition possibility described by (2.2). In this case, the task t_i represents the current task, and the next task is denoted by t_{next}. Later, in Subsection 3.2, it will be explained how to adapt this equation to our communication process, and we will provide a more detailed description of the selection process, given by Algorithm 1.

(3) *Go to the target task*: In this step, the robot goes towards the target task selected in the previous step (t_{next}). As was already mentioned, for the sake of simplicity, we will suppose that the time required by the robots to transit from one task to the next one is negligible; therefore, it has not been taken into account.

(4) *Communication of the next task*: In this step, the robot communicates to its teammates the new task chosen as the target. If there is no communication between robots, i.e., $MAX_{RANGE} = 0$ (notice that the parameter MAX_{RANGE} indicates the distance or range within which the robot is able to receive messages from the other robots; see Figure 2 for illustration), the robots skip this step and start the execution of the target task by returning to Step 1. Thus, when a robot arrives at a task, it broadcasts a message to the other robots with this information. The robot will repeat this communication process during the whole execution of the task, during which time no other task is selected by the robot. Due to the typical communication restrictions that exist in swarm systems, the robots are only able to send messages within a maximum range or distance, that, as explained before, we denote by MAX_{RANGE} (look at Figure 2). It must be highlighted that this is a very simple communication process, similar to a ping message, which does not imply any communication protocol. However, the simplicity of this mechanism is compatible with the required simplicity of any swarm system.

As the robot performs the above steps, it also receives, in parallel, the messages from the other robots, computing the number of robots allocated to each task, as can be observed in the communication processing flowchart of Figure 1. The communication processing scheme performs a loop while the simulation is running, where each iteration requires a time unit to be executed. Specifically, the following sequential steps are involved:

(1) The robot initializes to 0 the number of robots allocated to the tasks, where nr_j stands for the number of robots allocated to the task t_j and nt is the total number of tasks.

(2) The robot waits to receive messages from other robots, i.e., it receives the messages that the other robots have broadcast. It is important to remember that a robot can only receive messages from robots which are within the range MAX_{RANGE}.

(3) The robot updates the number of robots assigned to each task. This information will be used by the robot during the second step of the task processing process (see the task execution flow chart shown in Figure 1). This process returns a vector $NTA = \{nr_1, \ldots, nr_{nt}\}$, where each element nr_j denotes the amount of robots allocated to a given task t_j.

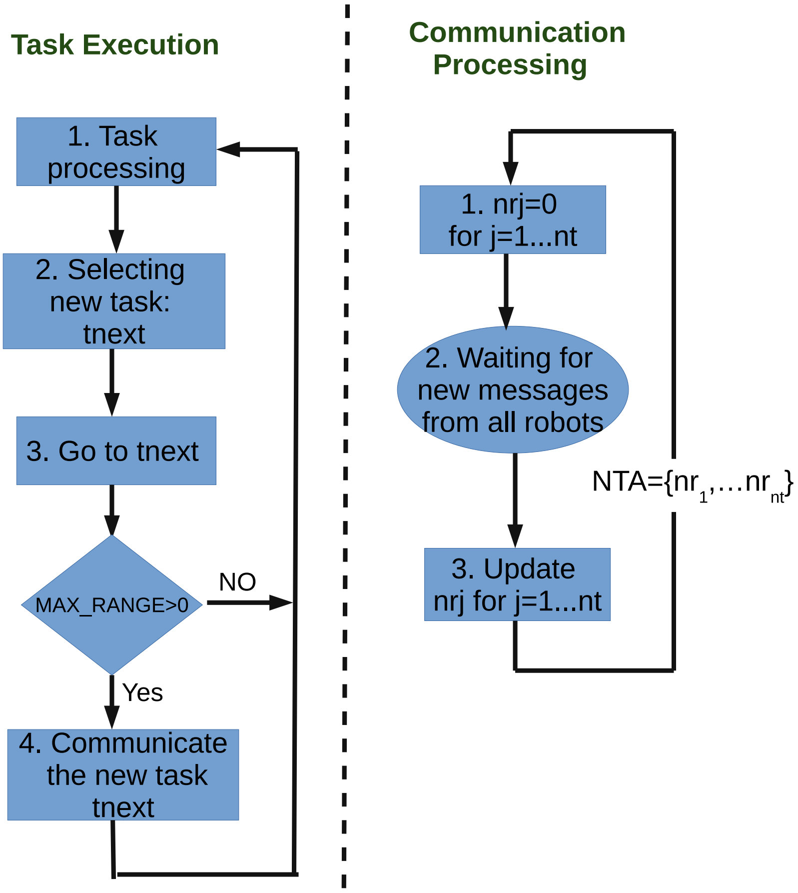
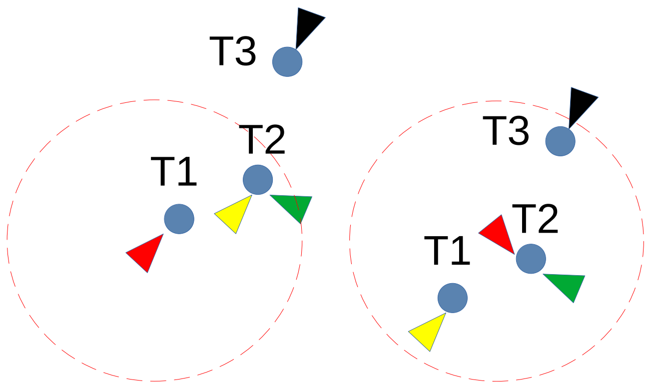


Figure 1. Steps followed by a robot during the execution of a task and during the communication process.



(a) Initial step.   (b) Next step.

Figure 2. Example of task execution with four robots and three tasks.

3.2. The selection of a new task

In this subsection, we provide a description about how each robot selects a new task to be executed as exposed in the second step of the task processing process in Section 3.1. To this end, we need to introduce how to update (2.2) which will be used by the robot in order to make such a decision. The more robots allocated to a task, the lower the possibility of transitioning to this task. The asymmetric distance Q_{Φ,q,d_E} introduced in (2.3) will now depend on the Euclidean distance $d_{ij} = d_E((x_i, y_i), (x_j, y_j))$ and, also, on the number of robots allocated to each task, as follows:

$$Q_{\Phi,q_n,d_E}\Big((nr_i, x_i, y_i), (nr_j, x_j, y_j)\Big) = (d_{ij} + \omega_w \cdot q_n(nr_i, nr_j))^n, \tag{3.1}$$

where $\Phi : (\mathbb{R}^+)^2 \to \mathbb{R}^+$ is the function given by $\Phi(x, y) = x + \omega_w \cdot y$ (the meaning of the parameter ω_w is explained below), $q_n(nr_i, nr_j) = max\{nr_j - nr_i, 0\}$ and nr_j, nr_i stand for the number of robots allocated to the task t_j and t_i, respectively

In light of Eq (3.1), (2.2) is rewritten in the following way:

$$p(r_k, ij) = \frac{\theta_{r_k}^n}{\theta_{r_k}^n + Q_{\Phi,q_n,d_E}^n\Big((nr_i, x_i, y_i), (nr_j, x_j, y_j)\Big)}, \tag{3.2}$$

where θ_{r_k} is the threshold constant for each robot r_k and ω_w denotes the weighting of the number of robots with respect to the distance between tasks and, thus, points out the importance of the aforesaid number of robots is with respect to the distance. Hence, low values of ω_w imply that a robot will select tasks close to it, instead of tasks with a high number of robots. Moreover, the number of robots and the distance are normalised by the ω_w parameter. Notice that both parameters play a similar role to those used in (2.2).

After dividing the numerator and the denominator of (2.2) by θ_{r_k}, we obtain the following equation:

$$p(r_k, ij) = \frac{1}{1 + \frac{1}{\theta_{r_k}^n} \cdot (d_{ij} + \omega_w \cdot q_n(nr_i, nr_j))^n}, \tag{3.3}$$

The same arguments as those given in Section 2 remain valid to show that the values of $p(r_k, ij)$ given by (3.3) are possibilities provided by fuzzy preorders that separate points when Dombi t-norms $T_{Dom}^{\frac{1}{n}}$ are under consideration.

As was described in Section 1, multiple robots can be assigned to each task and the possibility of transition from the current task (task t_i) to the next one (task t_j) is given by (3.3). Thus, a robot r_k, located at task t_i, will make the decision about the next task to carry out based on a fuzzy possibility distribution with the following values:

$$x_k = (p(r_k, i1), p(r_k, i2), ..., p(r_k, int)). \tag{3.4}$$

It must be noted that, in general, x_k does not satisfy the conditions of axioms of the probability distribution, as $\sum_{j=1}^{n} p(r_k, ij) \neq 1$; therefore, each element of x_k stands for a possibility of transition.

Figure 2 depicts an execution example of the above-mentioned process. There are three tasks, represented by the blue circles and labeled as $T1$, $T2$, and $T3$. There are also four robots marked by triangles of different colors (red, yellow, green and black). Initially (Figure 2(a)), the red robot is executing the task T1, the yellow and the green robots are executing the task T2, and the black robot is executing T3. The dashed red circle indicates the maximum communication range (MAX_{RANGE}) of the red robot. Thus, the red robot can get the information about the number of robots assigned to T2 which is two robots in this case. In contrast, T3 is out of the range of the red robot. The red robot, after executing Steps 2 and 3 of the task execution process (look at Figure 1) decides to go to the task T2, as can be seen in Figure 2(b). Meanwhile, the yellow robot has decided to go to the task T1. Now, the red robot is able to get the information about the number of robots assigned to each task (the three tasks), because all of them are within the communication range.

There are several decision making methods to choose the next task to be performed, and they are based on the activation of a possibility of transition by means of the generation of samples from a probability distribution induced by the possibilistic one (see [29]). In our case, with the aim of developing a robot decision making method, we use the interval method that can be described as follows:

(1) The x_k vector is normalized, dividing each element of the vector by the sum $\sum_{j=1}^{n} p(r_k, ij)$.

(2) Create n right-open intervals with width $\frac{p(r_k,ij)}{\sum_{j=1}^{n} p(r_k,ij)}$ ($j = 1, \ldots, nt$) over the interval [0, 1], that is

$$\left[0,\ \frac{p(r_k, i1)}{\sum_{j=1}^{nt} p(r_k, ij)}\right), \left[\frac{p(r_k, i1)}{\sum_{j=1}^{nt} p(r_k, ij)}, \frac{p(r_k, i1) + p(r_k, i2)}{\sum_{j=1}^{nt} p(r_k, ij)}\right), \ldots, \left[\frac{\sum_{j=1}^{nt-1} p(r_k, ij)}{\sum_{j=1}^{nt} p(r_k, ij)}, 1\right).$$

(3) Generate a sample u of a random variable $\mathcal{U}(0,\ 1)$.

(4) We will choose the task t_j whose index j satisfies the condition $u \in \left[\frac{\sum_{s=1}^{j-1} p(r_k,is)}{\sum_{j=1}^{nt} p(r_k,ij)}, \frac{\sum_{s=1}^{j} p(r_k,is)}{\sum_{j=1}^{nt} p(r_k,ij)}\right)$.

In light of the exposed facts, each robot will make a decision by following Algorithm 1 during Step 2 of the task execution process (see Figure 1).

Algorithm 1 Decision making for robot r_k

Require: Current task t_i
Require: $NTA = \{nr_1, \ldots nr_{nt}\}$
Require: θ_{r_l}, ω_w, $T = \{t_1, \ldots t_{nt}\}$
$x_k = \emptyset$
for $l = 1, \ldots, nt$ **do**
    $p(r_k, il) = \frac{1}{1 + \frac{1}{\theta_{r_l}^{n}} \cdot (d_{il} + \omega_w \cdot q_n(nr_i, nr_l))^n}$ (Equation (3.3))
    $x_k \leftarrow x_k \cup p(r_k, il)$
end for
$xNorm_k = Normalize(x_k)$
$next \leftarrow$ random integer from $xNorm_k$ with the interval method
return t_{next}

This algorithm has the following requirements for the input parameters: the current task assigned to the robot r_k, i.e., t_i; the θ_{r_l} and ω_w parameters defined in (3.3); the set of tasks T to calculate the distances between tasks; and, finally, the set $NTA = \{nr_1, \ldots nr_{nt}\}$, which contains the number of robots allocated to each task, where nr_k represents the number of robots assigned to the task t_k. The vector NTA is set during the communication processing process described in Figure 1. First, in line 1, the algorithm initializes the fuzzy set x_k to an empty set. Line 2 executes a loop where, for each task t_l, the transition possibility from the current task t_i to the task t_l is computed, denoted as $p(r_k, il)$ (see line 3). This value is calculated according to (3.3). In the last step of this loop, the value $p(r_k, il)$ is added to the fuzzy set x_k (line 4). Then, in line 6, the vector x_k is normalized as stated in the first step of the interval method described above. At last, line 7 computes the index $next \in \{1, \ldots, nt\}$ of the next task to execute according to Steps 3 and 4 of the aforementioned interval method. This index specifies the next task to be executed t_{next} that will be returned by the algorithm.

Note that, if there is no communication, i.e., $MAX_{RANGE} = 0$, the decision about the next task to execute only depends on the distance between the task and the robot (d_{ij} value in (3.3)). Indeed, the lack of communication is equivalent to the fact that all values of the vector NTA are zero; therefore, the number of robots allocated to a task has no impact on the decision making process. Consequently, the transition possibilistic function given by (3.5) is reduced to the following one:

$$p(r_k, ij) = \frac{1}{1 + \left(\frac{d_{ij}}{\theta_{r_k}}\right)^n}. \tag{3.5}$$

4. Experimental results

In this section, we present the simulations conducted to analyze the influence of communication on our task allocation mechanism. All experiments were performed by using Matlab, specifically release 2018a. Additionally, the source code for all developed scripts is publicly available in a repository located at this link*.

To validate the task allocation mechanism independently of other factors, we have made the following assumptions: (1) the travel time required by robots from one task to the next is zero; and (2) there are no collisions between robots. Two kinds of environments have been taken into account: environments with randomly-placed tasks and environments with tasks grouped in clusters, both of them being synthetic environments. For the first one, the tasks are located randomly, as can be observed in Figure 3. Figure 4 shows the location of the clustered tasks for the later experiments. As can be seen, the tasks are arranged into 2, 4, 6 and 8 groups or clusters. Each cluster of tasks is circled in red and labeled as C1,...,C8. For all experiments, 30 robots and 100 tasks have been used and the dimension of the environment is the same (width=600 units and height=600 units). The threshold value θ_{r_k} will always be, for all robots, equal to $\frac{d_{max}}{4}$, where d_{max} is the maximum distance between two tasks in the environments generated to carry out the experiments. In our case, d_{max} is equal to 800.5 units. It is important to mention that there is a relationship between d_{max} and the maximum range of communication: greater values of d_{max} will require greater values of MAX_{RANGE}, and vice versa. Due to this relationship, we have decided to keep constant d_{max} and to modify the MAX_{RANGE} value during the experiments. Since our purpose is to analyze the number of steps (time) required by the multi-robot system to visit, at least once, each task, tasks are never finished and therefore always remain in the same position from the beginning until the end of the simulation.

All results have been obtained after 500 iterations or steps. In order to keep the same initial

*https://github.com/joseGuerreroUIB/MultiComm conditions over all experiments, the number of tasks was always the same and they were always placed in the same position. Thus, the goal of the system was to decrease the number of simulation steps so that all tasks have been visited by at least one robot (which has been measured by applying the values of the parameter ω_w in (3.3)).

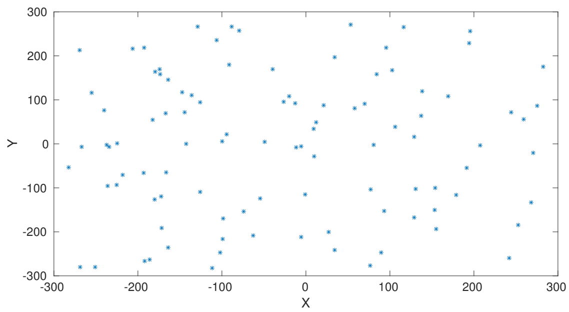
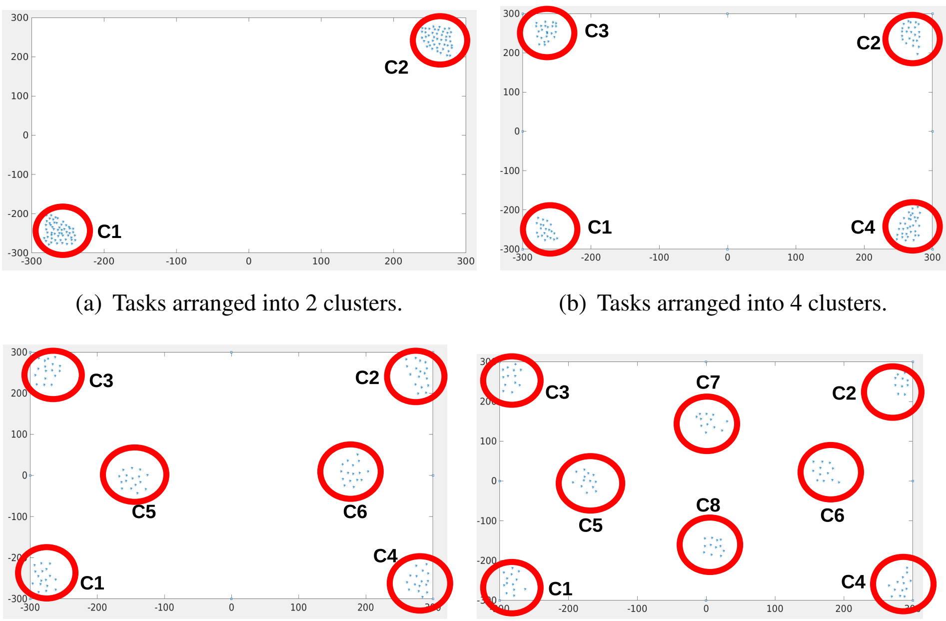


Figure 3. An example of an environment in which tasks are randomly placed. Blue dots represent the positions of the tasks.



(a) Tasks arranged into 2 clusters. (b) Tasks arranged into 4 clusters.

(c) Tasks arranged into 6 clusters. (d) Tasks arranged into 8 clusters.

Figure 4. Environments with 100 tasks arranged into the clusters used for the experiments. Blue dots represent the positions of the tasks. Each cluster is marked with a red circle and labeled as C1, ..., C8.

Next, we describe the results obtained from the experiments.

4.1. Experiments with randomly-placed tasks

Figure 5 shows the number of simulation steps that have been necessary for all tasks to be visited by a robot at least once when the tasks are placed randomly in the environment according to the parameter

ω_w in (3.3), running from $\omega_w = 0.2$ to $\omega_w = 2.0$. What is more, different values of the maximum communication range have been tested, from no communication strategies ($MAX_{RANGE} = 0$) to full communication between robots ($MAX_{RANGE} = 800.5$). Each experiment has been performed with 500 different random environments and the mean value of all of those experiments has also been depicted in Figure 5. As can be seen, the minimum number of simulation steps for all task visits is obtained when the maximum communication range is equal to 200 units and $\omega_w = 1.0$. In this case, the use of communication decreases by 7.5% the number of simulation steps relative to the strategy without communication. When the maximum communication range is high ($MAX_{RANGE} = 600$ or $MAX_{RANGE} = 800$), even when we use full communication between robots, there are no significant differences in terms of the results without communication. In contrast, it must be noted that for some values of MAX_{RANGE}, like for example when $MAX_{RANGE} = 400$, the communication, for some values of ω_w, increase the number of simulation steps.

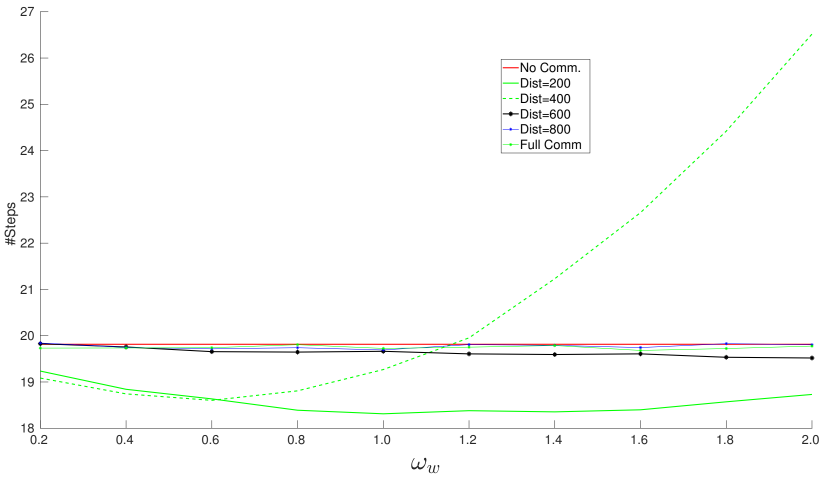


Figure 5. Number of simulation steps required to visit all tasks for random environments, considering different values of both ω_w and the maximum communication range (MAX_{RANGE}).

4.2. Experiments with clustered environments

Figures 6 and 7 show the number of simulation steps performed so that all tasks have been visited by at least one robot when the tasks are arranged into 2, 4, 6, and 8 clusters (see Figure 4). The ω_w parameter ran from $\omega_w = 0.2$ to $\omega_w = 2.0$. In this case, we fixed the maximum communication range and analyzed the impact of the parameter ω_w when different clusters are proposed. Hence, the maximum communication range MAX_{RANGE} is 200 units in Figure 6 and 400 units in Figure 7.

When $MAX_{RANGE} = 200$ units, the number of simulation steps decreases with increments of the values of ω_w. This behavior is seen more clearly when the number of clusters is low, like for instance with only 2 clusters. The minimum number of simulation steps carried out to visit all tasks is obtained when $\omega_w = 1.2$ with 6 clusters.

With $MAX_{RANGE} = 400$ units, a tendency similar to the one observed for randomly-placed task environments. In this case, when there are a lot of clusters (i.e., the number of clusters is equal to either 6 or 8) and the communication has a high weight value in the decision process (high values of the parameter ω_w), the communication process reduces the performance of the system. In contrast, when there is a low number of clusters (either 2 or 4 clusters), then the system behavior is similar to that exposed in Figure 6. The minimum number of simulation steps carried to visit all tasks is obtained when $\omega_w = 1.6$ with 4 clusters.

These results also demonstrate that, with 2 clusters, the number of simulation steps is, in some cases, higher than with more than two. This behavior is due to the low value of the probability transition function between these two clusters. In contrast, with in-between clusters (four or more), the robots can jump to closer clusters in order to perform faraway tasks.

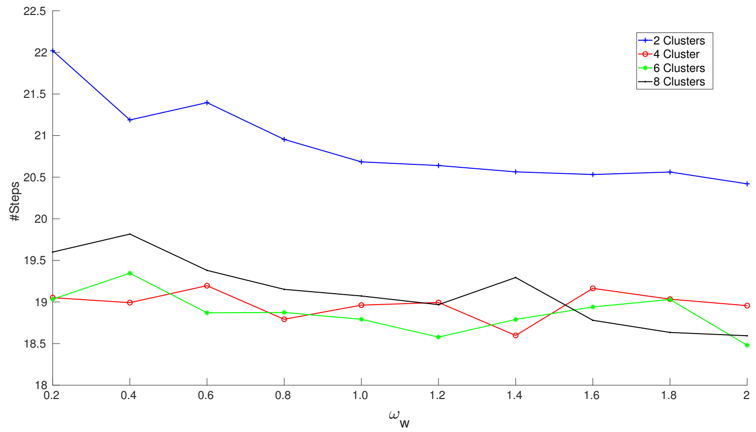


Figure 6. Number of simulation steps required to visit all tasks with different clusters, and for different values of ω_w. $MAX_{RANGE} = 200$ units.

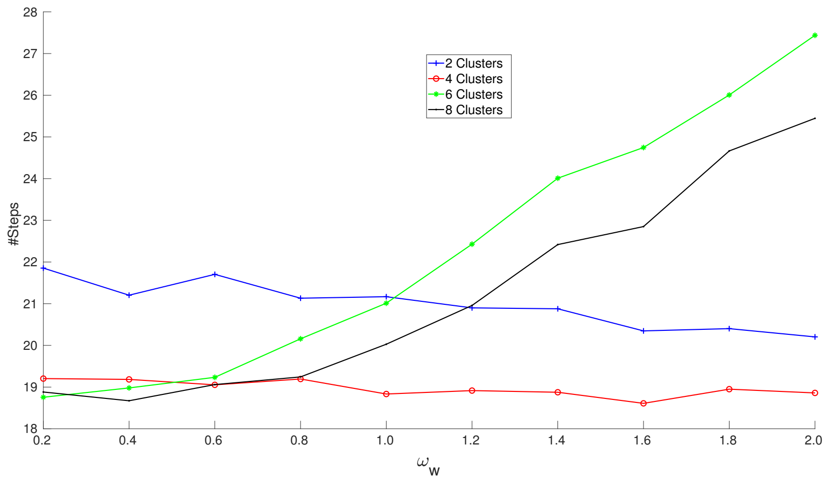


Figure 7. Number of simulation steps required to visit all tasks with different clusters, and for different values of ω_w. $MAX_{RANGE} = 400$ units.

Figure 8 depicts the total distance traveled by the robots without using communication when the tasks are clustered in 2, 4, 6 and 8 groups. As can be observed, the greater the number of clusters, the longer the total distance. When there are more clusters, for example 8, the robots must constantly transit from one group of tasks to another one, increasing the traveled distance. On the contrary, with a lower number of clusters, for example 2, the robots remain in their current cluster, because the distance between clusters is longer and the possibility for transiting between them is, in consequence, lower.

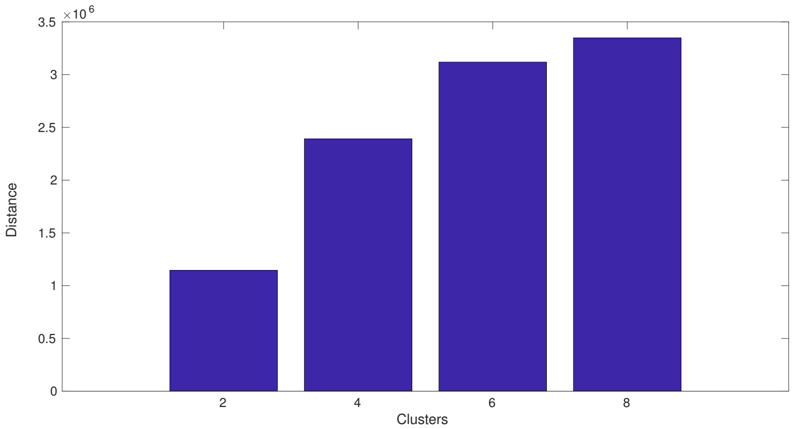


Figure 8. Total distance traveled by all robots with different clusters of tasks and without communication ($MAX_{RANGE} = 0$).

Figures 9 and 10 represent the total distance traveled by the robots with different numbers of clusters, different values of the parameter ω_w and with $MAX_{RANGE} = 200$ units and $MAX_{RANGE} = 400$ units, respectively. Both figures show the same behavior as that without communication, that is to say, the environments with a higher number of clusters imply longer distances. Besides, in both cases, and more specifically with $MAX_{RANGE} = 400$, the total traveled distance decreases with respect to the parameter ω_w. Remember that ω_w weights the number of robots with respect to the distance between tasks; therefore, if the value of ω_w is higher then the transition possibility to distant clusters is lower.

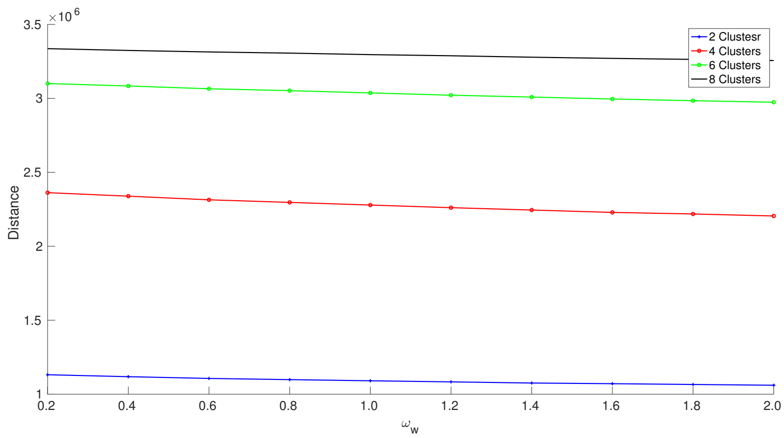


Figure 9. Total distance traveled by all robots with different clusters of tasks, and for different values of ω_w. $MAX_{RANGE} = 200$ units.

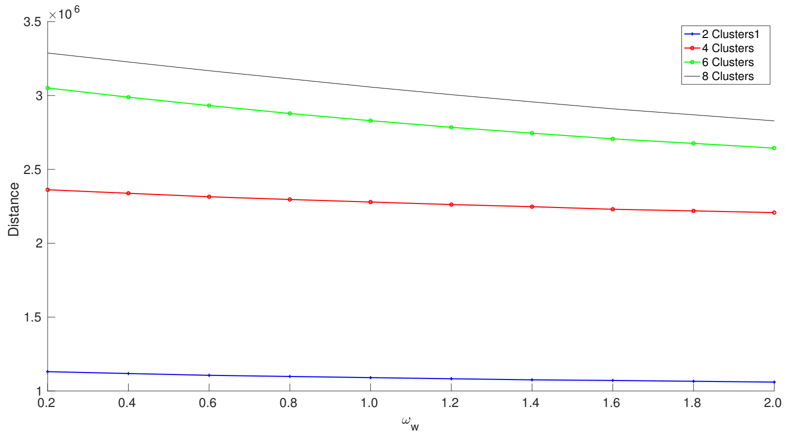


Figure 10. Total distance traveled by all robots with different clusters of tasks, and for different values of ω_w. MAX_{RANGE} = 400 units.

In order to conduct a deeper study of the system's behavior, we constructed Figures 11–14 to show the number of simulation steps required to visit all tasks when those tasks are arranged into 2, 4, 6, and 8 clusters as illustrated in Figure 4. Different values of MAX_{RANGE} and ω_w have been considered. Figure 11 depicts the results for 2 clusters. As can be seen, the use of a full communication strategy provides the worst results (the greatest number of simulation steps). In contrast, when a limited communication range is considered, even if $MAX_{RANGE} = 0$, the number of simulation steps decreases as the weight ω_w is increased. Furthermore, for the remaining number of clusters (look at Figures 12–14), as the number of clusters increases, there is a MAX_{RANGE} value that abruptly increases the number of simulation steps. From now on, the worst value of MAX_{RANGE} will be denoted as MAX_{WORST}. The simulations demonstrate that there is a correlation between the number of clusters and, therefore, the distance between them and the MAX_{WORST} value: The shorter the distance between clusters (or the greater the number of clusters), the lower the MAX_{WORST} value. Thus, when the number of clusters is equal to 4, MAX_{WORST} is equal to 600 (see Figure 12); when the number of clusters is equal to 6, MAX_{WORST} is equal to 400 (see Figure 13); finally, regarding the maximum number of clusters, clearly, the worst communication range is equal to 400 (see Figure 14). In light of these results, we can conclude that increasing communication (communication range increases) can increase the number of simulation steps; therefore, the system's performance degrades. These results are consistent with those obtained by Talamali et al. in [11] where, as was explained in Section 1, increasing the communication range could degrade the system's performance. It should be noted that the communication mechanism is very simple because we want to meet the requirements of swarm systems. These kinds of situations in which more information does not imply better results have been suggested by other authors using swarm-like mechanisms and remain a hot topic in research, as Talamali et al. pointed out.

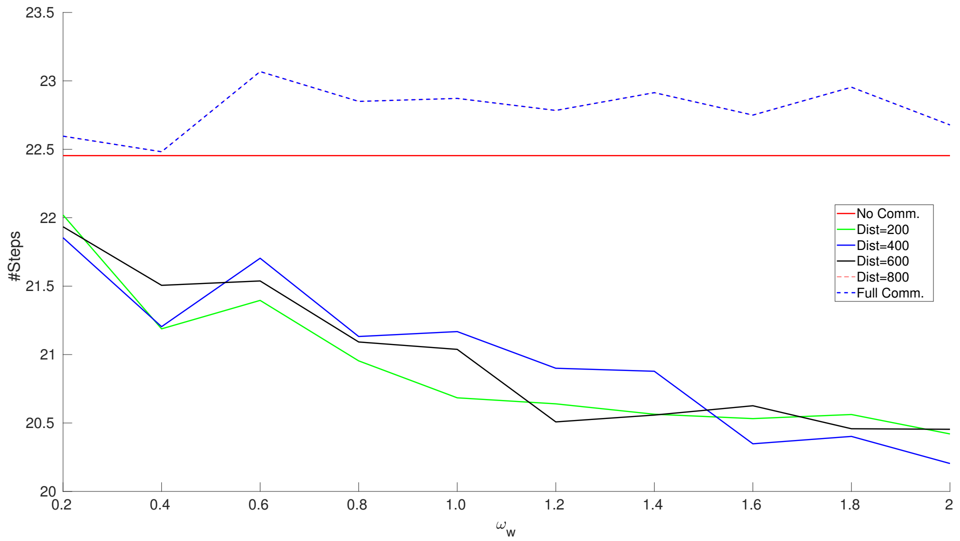


Figure 11. Number of simulation steps required to visit all tasks with 2 clusters, different values of MAX_{RANGE}, and different values of ω_w.

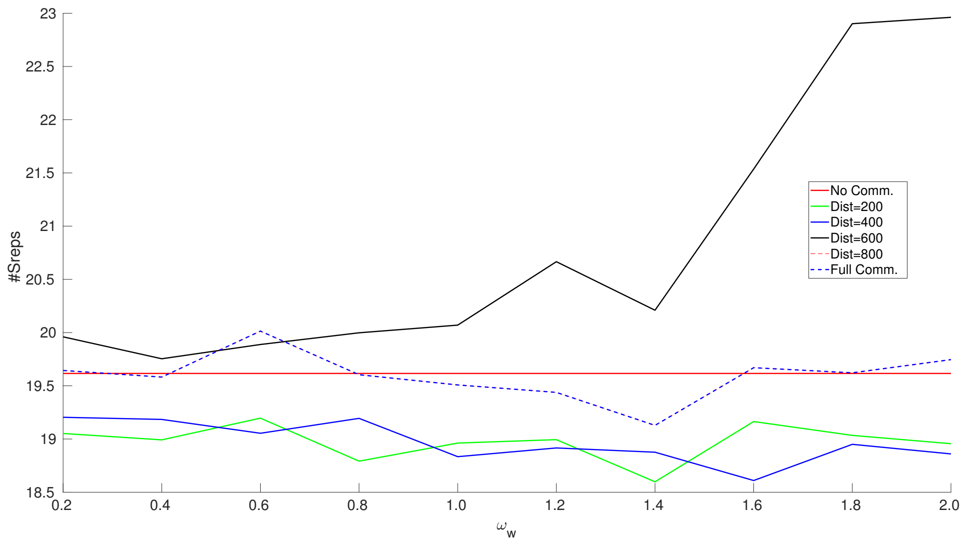


Figure 12. Number of simulation steps required to visit all tasks with 4 clusters, different values of MAX_{RANGE}, and different values of ω_w.

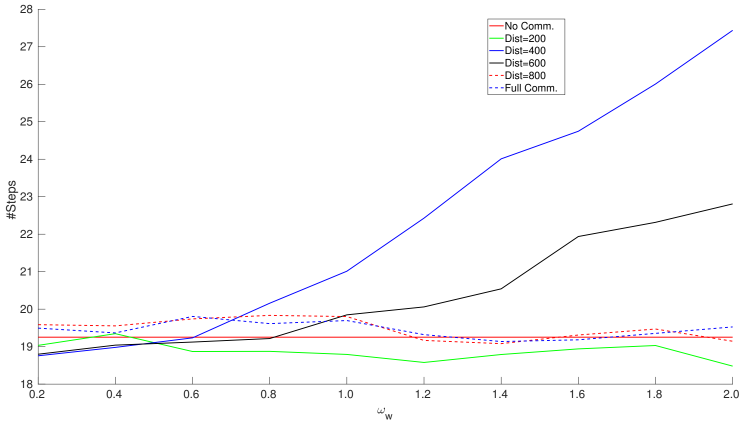


Figure 13. Number of simulation steps required to visit all tasks with 6 clusters, different values of MAX_{RANGE}, and different values of ω_w.

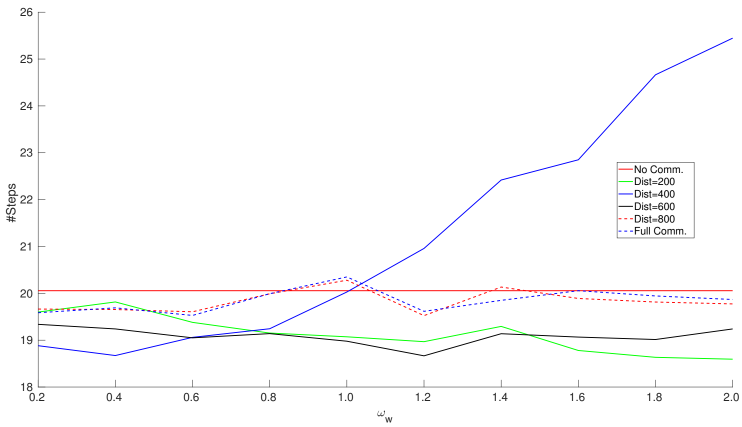
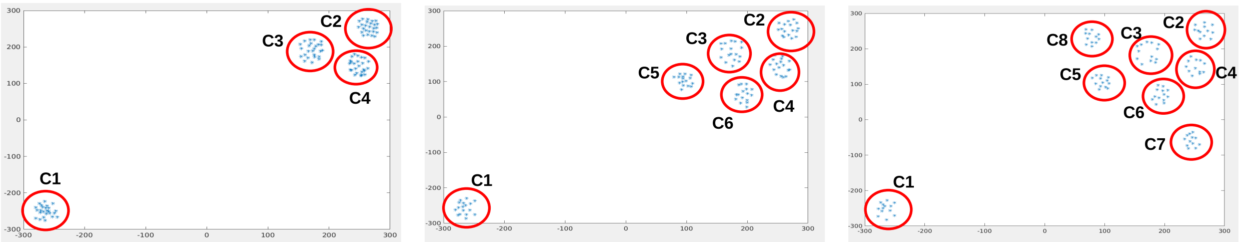


Figure 14. Number of simulation steps (Steps) required to visit all tasks with 8 clusters, different values of MAX_{RANGE}, and different values of ω_w.

The last set of simulations carried out to validate our approach considers the impact of the cluster distribution on the system's performance. Thus, instead of using the aforementioned clustered environments, which were arranged symmetrically as illustrated in Figure 4, clusters arranged asymmetrically are now under consideration. Figure 15 shows the three considered asymmetric clustered environments in which tasks are arranged into 4, 6, and 8 clusters. As can be observed, the clusters are not uniformly distributed over the environment; moreover, most of them are positioned in the upper right corner of the environment. The remaining parameters had the same values as in the previous experiments, i.e., the number of tasks was equal to 100, there were 30 robots, and d_{max} was equal to 800.5 units of distance.



(a) Tasks arranged into 4 clusters. (b) Tasks arranged into 6 clusters. (c) Tasks arranged into 8 clusters.

Figure 15. Environments with 100 tasks arranged into asymmetric clusters used for the experiments. Blue dots represent the positions of the tasks. Each cluster is marked with a red circle and labeled as C1, ..., C8.

Table 1 summarizes the number of simulation steps required to visit all tasks with the 4, 6, and 8 asymmetric clusters without using communication. This means that robots only take into account the distance to choose the next task to visit, i.e., the possibilistic transitions are given by (3.5). As can be seen, despite the results being very similar for all clusters, the number of simulation steps slightly increases when the number of clusters increases. Notice that these results are similar to those obtained with symmetric clusters of tasks.

Table 1. Number of simulation steps required to visit all tasks with different amounts of asymmetric clusters without using communication.

-	Steps
4 Clusters	18.72
6 Clusters	19.09
8 Clusters	19.22

Figures 16 and 17 show the mean number of simulation steps required to visit all tasks with 4, 6 and 8 asymmetric clusters (look at Figure 15). The maximum communication range has been fixed and different values of the parameter ω_w have been tested. The obtained results are equivalent to those represented in Figures 6 and 7, but with asymmetric clusters instead.

Figure 16 depicts these results when the parameter MAX_{RANGE} is equal to 200 units. As can be seen, for any number of clusters, if ω_w is lower than 0.8, the number of simulation steps is slightly lower than the corresponding results without communication. For instance, if $\omega_w = 0.2$, then the mean number of simulation steps with communication and 8 clusters is equal to 18.14, but without communication (refer to Table 1), this value is equal to 19.22. What is more, the results obtained for low values of ω_w and these asymmetric clusters are similar to the results without asymmetry, as can be observed in Figure 6. It seems that the distribution of tasks does not impact the system's performance when the communication does not have a high value of ω_w. The environments with 2 clusters, as shown in Figure 6, have the worst results. For values of ω_w higher than 0.8 and 4 clusters, the number of simulation steps is very similar to those needed when 2 clusters without asymmetrical task arrangement are considered. This is due to the fact that the clusters of tasks $C2$, $C3$, and $C4$ in the asymmetric environment (see Figure 15) are so close to each other that all of them can be understood to be one virtual, unique, bigger cluster. The number of simulation steps increases significantly according to the parameter ω_w when ω_w is greater than 0.8 and we consider 6 and 8 clusters. The exposed results are very similar to those obtained without asymmetric clusters and with MAX_{RANGE} = 400 (see Figure 7). The reason for this may be the fact that the communication range is too short for robots to cover clusters from $C2$ to $C8$ when the tasks are asymmetrical.

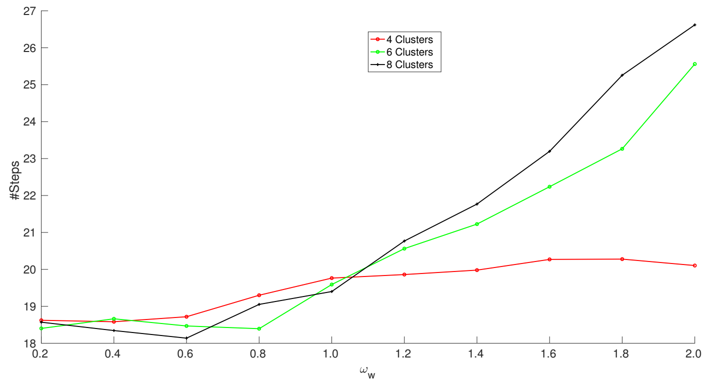


Figure 16. Mean number of simulation steps required to visit all tasks with different amounts of asymmetric clusters, and for different values of ω_w. MAX_{RANGE} = 200 units.

It remains for us discuss the system's behavior when MAX_{RANGE} = 400. Figure 17 shows the obtained results when the parameter MAX_{RANGE} is equal to 400 units for different amounts of asymmetric clusters and different values of the parameter ω_w. As can be observed, the number of simulation steps decreases for the experiments with MAX_{RANGE} = 200 and asymmetrical task arrangement. Moreover, the aforesaid number of simulation steps does not increase according to ω_w as significantly as it did in the aforementioned asymmetric case with MAX_{RANGE} = 200. Furthermore, the system behaves in a more stable way in relation to the parameter ω_w when higher values of MAX_{RANGE} are considered. It is worth noting that with MAX_{RANGE} = 400, the communication range is able to cover the entire cluster area from $C2$ to $C4$ (look at Figure 15); therefore, robots can make their decisions with less uncertainty than with MAX_{RANGE} = 200. What is more, the numbers of simulation steps for the highest values of ω_w with asymmetric task arrangement are very similar to those showed for 2 clusters without asymmetry in Figure 7. Thus, the behavior of a system with 8 asymmetric clusters, as they are arranged in Figure 15(c), is similar to the behavior of a system with 2 clusters without asymmetric arrangement, as in Figure 4(a). Notice that these results are similar to those mentioned above with MAX_{RANGE} = 200 and are due to the fact that the clusters from $C2$ to $C8$ are so close that they can be seen as a single cluster by our system.

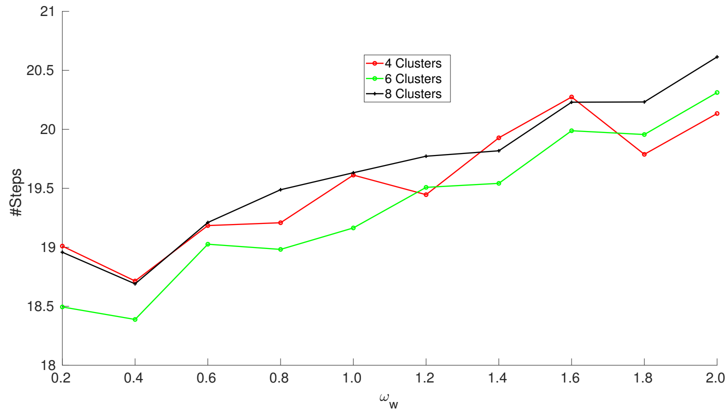


Figure 17. Mean number of simulation steps required to visit all tasks with different amounts of asymmetric clusters, and for different values of ω_w. MAX_{RANGE} = 400 units.

Finally, Figures 18 and 19 illustrate the standard deviation (σ) of the number of simulation steps required to visit all tasks with 4, 6, and 8 asymmetric clusters, i.e., these figures represent the standard deviation of the results depicted in Figures 16 and 17, respectively. Upon comparing Figures 18 and 19, we can observe that the standard deviation is generally smaller when MAX_{RANGE} is equal to 400 units than the corresponding results obtained with MAX_{RANGE} = 200. More to the point, with MAX_{RANGE} = 400, the maximum standard deviation across all clusters is 5.07 units, whereas the global maximum value of σ is 9.3 units when MAX_{RANGE} = 200. When MAX_{RANGE} = 200, in general, larger values of the ω_W parameter result in larger values of σ; additionally, it is important to note that the standard deviation is also highly influenced by the number of clusters; specifically, when the number of clusters is either 6 or 8, σ experiences a sudden increase when the ω_W parameter exceeds 1.0. In contrast, when MAX_{RANGE} = 400, the standard deviation remains more stable in terms of variations in the ω_W parameter. This fact could suggest that effective communication contributes to more robust system behavior.

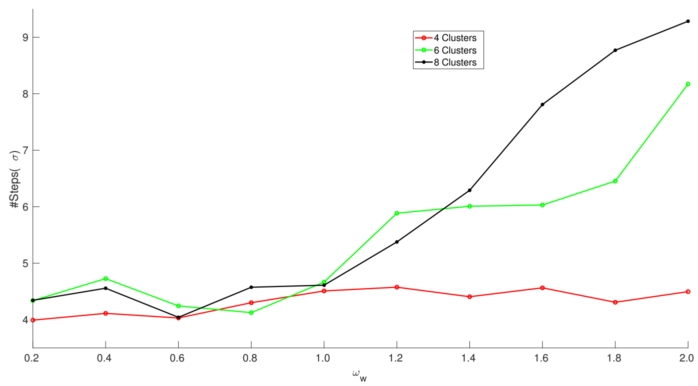


Figure 18. Standard deviation (σ) of the number of simulation steps required to visit all tasks with different amounts of asymmetric clusters, and for different values of ω_w. MAX_{RANGE} = 200 units.

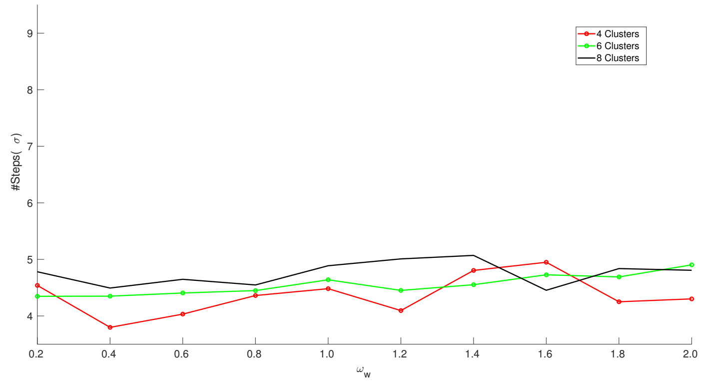


Figure 19. Standard deviation (σ) of the number of simulation steps required to visit all tasks with different amounts of asymmetric clusters, and for different values of ω_w. MAX_{RANGE} = 400 units.

5. Conclusions and future work

In this paper, we have studied how to model a multi-robot task-allocation process by using fuzzy preorders as transitions when a simple communication mechanism between robots is under consideration. Thus, we have extended our previous work by including a communication mechanism that allows one to share the information about the number of robots allocated to each task, in a response threshold-like algorithm. This new information allows us to model the transitions through the use of appropriate fuzzy preorders. The simulation results, under Matlab, have demonstrated the impact of both the maximum communication range (MAX_{RANGE}) and the parameter ω_w on the system's performance. The use of the parameter ω_w helps us to weight the number of robots with respect to the distance between tasks and, thus, determines the importance of the aforementioned number of robots according to the distance. The simulations have been performed for different kinds of environments. In light of the simulation results obtained, the following conclusions can be drawn:

- For specific combinations of MAX_{RANGE} and values of ω_w, our approach enhances the system's performance by reducing the number of simulation steps relative to the case without communication. Nevertheless, in general, as discussed in Section 4, a better communication mechanism with a larger MAX_{RANGE} does not necessarily lead to improved results; in fact, in some cases, it may even decrease the system's performance. Furthermore, there appears to be a correlation between the distribution of tasks and the best MAX_{RANGE} value.
- The use of the proposed communication mechanism not only improves the system's performance in terms of simulation steps for the same system without communication, but it also benefits from its simplicity, making it applicable to any swarm system.
- Overall, larger values of MAX_{RANGE} tend to result in lower standard deviations. Thus, the communication mechanism clearly contributes to the realization of more stable and robust outcomes.

It is worth emphasizing that this paper introduces, for the first time, a formalism based on fuzzy preorders, asymmetric distances and aggregation to model multi-robot communication. This novel formalism is able to automatically generate new transition functions from previous ones through the use of fuzzy preorders. Some realistic applications of this response threshold algorithm could be those where the task should be visited as soon as possible for the first time to check its initial status. Besides, the status of each task should be periodically checked to monitor its evolution. One specific type of task that aligns with all of these characteristics is the use of nanorobots in cancer detection and treatment [30]. Initially, the nanorobots must visit the potential tumors. Subsequently, they should revisit these regions over time to track their evolution and verify treatment correctness. In this context, the clusters depicted in Figure 4 could represent the potential tumor-affected regions.

In a future work, a deeper study of the impact of ω_w will be carried out. With this aim, we expect to include a machine learning algorithm to fit the parameters of the proposed task-allocation method. Furthermore, we expect to implement and test this method in a colony of real robots.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was part of projects PID2022-139248NB-I00 and PGC2018-095709-B-C21 funded by MCIN/AEI/10.13039/501100011033 and "ERDF A way of making Europe" and from project BUGWRIGHT2. This last project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 871260. This publication reflects only the authors views and the European Union is not liable for any use that may be made of the information contained herein.

Conflict of interest

The authors declare no conflict of interest in this paper.

References

1. G. Theraulaz, E. Bonabeau, J. N. Denuebourg, Response threshold reinforcements and division of labour in insect societies, *Proc. Roy. Soc. Lond. B: Biol. Sci.*, **265** (1998), 327–332. https://doi.org/10.1098/rspb.1998.0299
2. W. Agassounon, A. Martinoli, *Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems*, In AAMAS'12, Bolonia, Italy, 2002, 1090–1097.
3. B. Heap, M. Pagnucco, Repeated sequential single-cluster auctions with dynamic tasks for multi-robot task allocation with pickup and delivery, *Lect. Notes Comput. Sci.*, **8076** (2013), 87–100. https://doi.org/10.1007/978-3-642-40776-5_10
4. M. Otte, M. J. Kuhlman, D. Sofge, Auctions for multi-robot task allocation in communication limited environments, *Auton. Robot.*, **44** (2020), 547–584. https://doi.org/10.1007/s10514-019-09828-5

5. R. J. Marcotte, X. P. Wang, D. Mehta, E. Olson, Optimizing multi-robot communication under bandwidth constraints, *Auton. Robot.*, **44** (2020), 43–55. https://doi.org/10.1007/s10514-019-09849-0

6. A. Contini, A. Farinelli, Coordination approaches for multi-item pickup and delivery in logistic scenarios, *Robot. Auton. Syst.*, **146** (2021), 103871.

7. S. Chowdhury, E. Sklar, *Investigating the impact of communication quality on evolving populations of artificial life agents*, In: the 13th European Conference on Artificial Life, 2015. https://doi.org/10.7551/978-0-262-33027-5-ch096

8. A. M. Deshpande, R. Kumar, M. Radmanesh, N. Veerabhadrappa, M. Kumar, A. A. Minai, *Self-organized circle formation around an unknown target by a multi-robot swarm using a local communication strategy*, In 2018 Annual American Control Conference (ACC), 2018, 4409–4413.

9. M. Dorigo, *Swarm-bot: An experiment in swarm robotics*, In Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005, 192–200.

10. D. A. Patil, M. Y. Upadhye, F. S. Kazi, N. M. Singh, *Multi robot communication and target tracking system with controller design and implementation of swarm robot using arduino*, In 2015 International Conference on Industrial Instrumentation and Control (ICIC), 2015, 412–416.

11. M. S. Talamali, A. Saha, J. A. R. Marshall, A. Reina, When less is more: Robot swarms adapt better to changes with constrained communication, *Sci. Robot.*, **6** (2021), 1–14. https://doi.org/10.1126/scirobotics.abf1416

12. J. Gielis, A. Shankar, A. Prorok, A critical review of communications in multi-robot systems, *Curr. Robot. Rep.*, **3** (2022), 213–225. https://doi.org/10.1007/s43154-022-00090-9

13. J. Guerrero, O. Valero, G. Oliver, Toward a possibilistic swarm multi-robot task allocation: Theoretical and experimental results, *Neural Process. Lett.*, **46** (2017), 881–897. https://doi.org/10.1007/s11063-017-9647-x

14. L. A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Set. Syst.*, **3** (1971), 177–200. https://doi.org/10.1016/S0165-0114(99)80004-9

15. H. Prade, D. Dubois, *Fuzzy sets and systems: Theory and applications*, Academic Press, 1980.

16. K. Avrachenkov, E. Sanchez, Fuzzy markov chains and decision making, *Fuzzy Optim.Decis. Ma.*, **1** (2002), 143–159.

17. J. Recasens, *Indistinguishability operators: Modelling fuzzy equalities and fuzzy equivalence relations*, Springer, 2010.

18. J. Guerrero, J. Miñana, O. Valero, *On the use of fuzzy preorders in multi-robot task allocation problem*, In Cham Springer, editor, International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, 2018, 195–206.

19. G. Mayor, O. Valero, Aggregation of asymmetric distances in computer science, *Inform. Sci.*, **180** (2010), 803–812. https://doi.org/10.1016/j.ins.2009.06.020

20. E. Castello, T. Yamamoto, F. D. Libera, W. G. Liu, A. F. T. Winfield, Y. Nakamura, et al., Adaptive foraging for simulated and real robotic swarms: The dynamical response threshold approach, *Swarm Intell.*, **10** (2016), 1–31. https://doi.org/10.1007/s11721-015-0117-7

21. Y. M. Yang, C. J. Zhou, Y. T. Tin, *Swarm robots task allocation based on response threshold model*, In ICARA'09, Willengton, New Zeland, 2009, 171–176.
22. L. A. Zadeh, Similarity relations and fuzzy orderings, *Inform. Sci.*, **1** (1971), 177–200. https://doi.org/10.1016/S0020-0255(71)80005-1
23. R. P. Klement, R. Mesiar, E. Pap, *Triangular norms*, Kluwer Academic Publishers, 2000.
24. M. M. Deza, E. Deza, *Encyclopedia of distances*, Springer, 2009.
25. D. Dubois, H. Prade, S. Sandri, On possibility/probability transformations, *Fuzzy logic*, **12** (1993). https://doi.org/10.1007/978-94-011-2014-2_10
26. J. Harwell, M. Gini, *Swarm engineering through quantitative measurement of swarm robotic principles in a 10,000 robot swarm*, In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), 2019.
27. J. Guerrero, J. J. Miñana, O. Valero, G. Oliver, Indistinguishability operators applied to task allocation problems in multi-agent systems, *Appl. Sci.*, **7** (2017), 963. https://doi.org/10.3390/app7100963
28. M. D. M. Bibiloni-Femenias, J. Guerrero, J. J. Miñana, O. Valero, Indistinguishability operators via yager t-norms and their applications to swarm multi-agent task allocation, *Mathematics*, **2** (2021), 190. https://doi.org/10.3390/math9020190
29. S. Chanas, M. Nowakowski, Single value simulation of fuzzy variable, *Fuzzy Set. Syst.*, **25** (1988), 43–57. https://doi.org/10.1016/0165-0114(88)90098-X
30. M. Aggarwal, S. Kumar, The use of nanorobotics in the treatment therapy of cancer and its future aspects: A review, *Cureus*, **14** (2022), 1–8. https://doi.org/10.7759/cureus.29366

AIMS Press