



Surveillance task optimized by Evolutionary shared Tabu Inverted Ant Cellular Automata Model for swarm robotics navigation control



Hamilton J.M. Lopes ^{a,c,*}, Danielli A. Lima ^{a,b}

^a Instituto Federal do Triângulo Mineiro (IFTM) Campus Patrocínio, Minas Gerais, Brazil

^b Computer Science Department, Leader of Laboratório de Inteligência Computacional e Robótica (LICRO), Avenida Liria Teresinha Lassi

Capuano, 255, Bairro Univeritário, Patrocínio, Minas Gerais, Zip Code: 38747-792, Brazil

^c Computer Science Department, Scholarship of National Council for Scientific and Technological Development (CNPq), Laboratório de Inteligência Computacional e Robótica (LICRO), Avenida Liria Teresinha Lassi Capuano, 255, Bairro Univeritário, Patrocínio, Minas Gerais, Zip Code: 38747-792, Brazil

ARTICLE INFO

Keywords:

Cellular automata
Swarm robotics
Genetic algorithms
Surveillance
Dynamical systems
Tabu search

ABSTRACT

Swarm robotics is an area of research that has attracted several researchers. This field is part of the collective robotics approach that is inspired by the self-organized behaviors of social animals. From interactions between agents, guided by simple rules, it is possible to design emerging collective behaviors capable of performing complex tasks in an organized manner, for the coordination and control of a large number of robots. In this article, we proposed a new model that combines different techniques of natural and evolutionary computing: we mainly focus on ideas and concepts about cellular automata, social pedestrians behavior in evacuation, genetic algorithms, inverted pheromone from ant colonies and Tabu search, as hybrid search mechanism. The objective of this new model is to provide an advance of surrogate techniques for swarm robotics as a field of science and engineering and that may be relevant to deal with the robotic surveillance task. The new model is called Genetic Shared Tabu Inverted Ant Cellular Automata, or shortly GSTIACA. Initially, it was made a meta-optimization of the surrogate parameters through a genetic algorithm. Then, we apply these new parameters for hundreds steps in a robot control and navigation algorithm based on cellular automata techniques in different environments types. The system global search takes place at different times, one of which is based on the spread of pheromone in the environment and the other which is based on the memory sharing based on Tabu search. The novelty of this work is precisely the Tabu search application as a local and used as a shared global search algorithm. Besides that, we developed a new algorithm for swarm robotics that integrates different artificial intelligence techniques and natural computing not yet used all together in precursor works. In addition, we reduced the cost of processing the pheromone decline calculation using an asynchronous cellular-automaton. Later, we contrasted the new model in different situations, and saw that the new algorithm proposed here is better than its precursors. Finally, we did a test using the e-Puck architecture within the Webots simulation environment to prove that the mathematical model proposed herein is capable of being applied in the real world application.

1. Introduction

Swarm robotics is a multi-agent system coordination and control approach, where each agent is considered a physically simple robot [1]. From the cooperation and system stigmergy, a collective behavior can emerge from the interaction between robots and

* Corresponding author at: Instituto Federal do Triângulo Mineiro (IFTM) Campus Patrocínio, Minas Gerais, Brazil.

E-mail addresses: hamilton.hjml@gmail.com (H.J.M. Lopes), danielli@iftm.edu.br (D.A. Lima).

the environment [2,3]. This approach has been widely investigated by several researchers [4–7] due to the parallelism and multi-objective [8] that can occur in this system in relation to the traditional robotics usage [9], where a single robot can perform tasks sequentially [10,11]. The first swarm robotics approaches emerged in the artificial intelligence field, from the biological bees study [12], ants [4] and other populations where swarming behavior occurred [13,14]. Swarm robotics is inspired by the emerging behavior observed in social insects [15] or even pedestrians interactions [16], called swarm intelligence, but is not limited to this. The robotics research may have aspects in the robots design study in their physical structures in the control of their behaviors [17] and even in the tasks allocations applications [18,19], such as path planning [20,21], mining [22], search and rescue [23] and foraging [4]. Thus, relatively simple individual rules can generate a large set of complex patterns for robot swarm behavior coordination.

Based on this same context, elementary cellular automata (ECA) are the models of temporal states evolution with simple rules capable of learning [24] a complex global behavior [4,25]. There are other types of cellular automata (AC) that could be used in modeling complex systems such as fire spread [26], Covid-19 transmission [27], optimization [28], oil spread [29], pedestrian evacuation [30,31], cryptography models [32,33] or even for the generation of complex cellular automata [34]. In our work we use the 2D bidimensional CA, with square cells, probabilistic transition rule, hybrid update (synchronous and asynchronous), Moore neighborhood using radius 1r, and non-periodic contour. In our problem, the states can be: free, obstacle, robot, recently visited. In addition, cells have different probability of visit, depending on the amount of pheromone inserted in the cell. Therefore, it is very easy to accept the importance of this study, when trying to understand the complexity that can be applied across all natural and social sciences [35,36]. This approach being an interesting way to combine swarm control and robotics [37], as it is an area of collective and social interaction between robots. A decisive component in robotics is the interaction between a team of robots [23], thus allowing a high sociability among the agents [38]. Swarm behavior involves the constant change of individuals states in mutual cooperation [6], as well as the group behavior, making this approach interesting for robot control based on the evolution of these cellular automata in mobile robotics of multiple agents, since they are seen as dynamic systems [39].

Another interesting approach, which is also based on collective intelligence, is the ants behavior [40]. Ants generally walk aimlessly while exploring the environment in search of food until they return to the colony leaving a trail of pheromone [41]. From there, other ants can interact locally with these pheromones and from this situation, a coordinated global behavior emerges [42]. Another situation regarding the pheromone usage can be observed when ants are in a panic state, and in this case, they release a pheromone that makes them avoid places recently visited by other ants, that is, a repulsive force emerges in the collective system [41]. The more ants pass through the same place, more time it will take for the trail's pheromone to evaporate [40]. Pheromone evaporation appears exactly as a way of converging to an optimal local solution. This ant colony optimization (ACO) approach has been widely applied in robotics [17,43], due to the fact that ants make an excellent exploration and environment coverage [44], thus, they are conducive to the robotic surveillance [45,46] task, which will be investigated in our work. Additionally, the repulsive pheromone idea used in our work to increase the area of exploration by robots will be increased through the memory addition, based on Tabu Search (TS) [47,48], which is able to prevent recently visited areas from being visited again, thus improving the surveillance task performance [4].

From these approaches it is possible to build an algorithm for the robotic surveillance task. The mobile robotic surveillance task is capable of carrying out patrols in the areas to be protected, and communicating them to a central computer [49]. The robots have sensors and a web camera that allow remote rooms observation [50] through which they move. The system objective is to patrol the largest number of rooms in the shortest time, with an acceptable robots number [7]. For the control of this multi-agent system, two types of searches can be detailed: a global search and a local search. The global search is made from the agents indirect communication by the pheromone and by its evaporation [51]. The local search will be made by direct communication between agents according to their proximity within the CA range, exchanging messages asynchronously [52]. In addition, it is possible to glimpse the local search given by the agent's finite memory based on the Tabu search [47,48]. Therefore, this system has a considerable number of input parameters, which makes it difficult to optimally configure all of them [53]. The way in which the parameters are configured can influence the final result [54]. The optimization through different possibilities to automate these patterns, allows a significant improvement in the final algorithm performance [55].

In this work, to allow a wide range of experiments and applications, the new model parameters will be optimized by a Genetic Algorithm (GA), a search technique used in artificial intelligence [56], created to find approximate solutions in search and optimization problems, being a particular class of evolutionary algorithm [57]. This algorithm uses techniques inspired by evolutionary biology, techniques such as heredity, mutation, natural selection and recombination [58]. However, the fitness function is difficult to measure in order to optimize the number of complete environment patrols. That is, it is difficult to define a way to assess the performance or fitness of the object of study.

To solve this problem we use what is known as surrogate model [59]. Surrogate is a method used when a result of interest cannot be easily measured directly; therefore, the result model is used [60]. In our work, the surrogate can be understood in two ways: (1) the navigation model based on CA and its improvements, a parameter passed to calculate the GA fitness and (2) the complete model including GA meta-optimization. Since in this case we are dealing with an engineering control problem that requires experiments and simulations to evaluate the team control surveillance and constraint functions as a control function of variables. Besides that, we aim to mimic the simulation behavior model as closely as possible while being computationally cheap to evaluate, instead of using the real robotics. Most robotics problems require experiments and/or simulations to assess the objective functions and design constraints in terms of their variables [54]. For many real-world problems, however, a single simulation can take many minutes, hours or even days to complete [61]. Alternatively, a surrogate genetic algorithm (SGA) will be used to address the optimization of the multi-agent controller parameters for the robot patrol task [61–63]. In addition, an improvement in the

pheromone decline update, proposed here, will be treated as a way of minimizing the surrogate execution time for robot navigation, making the simulation computationally cheaper. In this work the novelty is the usage of asynchronous cellular automata cells update in the decline pheromone process.¹ A new navigation model with several robots is proposed for surveillance tasks. Our model was named Genetic Shared Tabu Inverted Ant Cellular Automata model, or just GSTIACA model. The GSTIACA model is described in the following sections and the results are demonstrated to exemplify the proposed model features. This article is structured as follows: Section 2 relates the state of the art accordingly to literature review. Section 3 describes the proposed surveillance task model using natural computing triumph strategies: cellular automata, genetic algorithms, ant colonies optimization, shared tabu memory for local and global swarm communication, that will be detailed in the next sections. Section 4 shows the simulation results for the model refinement and parameters optimization of this hybrid model using a genetic algorithm. It also presents some experiments to show the best environment configuration, test and measure the final model with the optimized parameters, measuring steps quantities and pheromone distribution. The results discussion of all the proposed model is shown in Section 5. Section 6 concludes the article and suggests future work, doing a comparison between main results and state of the art.

2. Background

This section is divided into two subsections, the first one brings the evolution and state of the art of all robot navigation models based on cellular automata for the robotic surveillance task. Although there are other models in the robot navigation literature, in this section we seek to detail only the robot navigation models for the robotic surveillance task, without considering the CA-based models for the path-planning [20,64], robotics mimeting pedestrian behavior [65], foraging task [4,44,66–68] and garbage collection task [17]. In the other subsection, we will present, in a summarized and comprehensive way, some of the ideas that led us to create a new CA-based model meta-optimized via GA for the foraging task. Details will be explained in the section describing the model, as well as the respective results.

2.1. State of art

In this subsection, we will present the state of the art regarding robot navigation models based on two-dimensional cellular automata and how each of them evolved. In addition, we will present the characteristics of each of the models in a summarized way (see Fig. 1).

The first model for swarm robotics created within the context of the surveillance task was the Inverted Ant Cellular Automata model, or shortly IACA, created by Lima et al. in 2016 [69]. In that work, four parameters were adopted. However, for comparison purposes, only the decay effect was alternated empirically as a comparison effect with the generic model of [6], a navigation model of surveillance robots that did not make use of cellular automata. In 2017, Tinoco, Lima and Oliveira [49] improved the IACA precursor model in order to add some parameters that would refine the real experiment. In this sense, the new model created was called Inverted Ant Cellular Automata with Discrete pheromone diffusion and Inertial motion, or shortly IACA-DI. The GIACA (Genetic Inverted Ant Cellular Automata) model implemented here also brings an improvement of the IACA optimized by a non-random GA, guided by biological and evolutionary strategies presented in [70].

In addition, other works by Oliveira, Tinoco and collaborators were created in order to improve the performance of the IACA precursor model, among them we can cite the 2018 work [71], in which the authors sought to study five different approaches for moving robots, including a totally random approach, which was discarded for not showing good results, an elitist approach and a strategy they call pure, which had intermediate results. The best performing strategies were the deterministic strategy considering its higher number of completed task and the elitist decision, which returned the most homogeneous coverage. Another strategy investigated in 2019 [72] uses a heterogeneous team of robots to perform the robotic task. A very promising evolution of the work was the adoption of a genetic algorithm for the refinement of 4 parameters of the model, as presented in the article [73]. This strategy of adopting a GA as a meta-optimizer was also worked on in the article by Lopes and Lima in 2021 [74]. In this article, the model was called GIACADI, Evolutionary with Genetic algorithm for Inverted Ant Cellular Automata Discrete Inertia, in order to refine all model parameters, not just 4, as in other work by Tinoco et al. The GA used in this work uses a different fitness function as well as a different selection form, in which only 1 single child is generated per crossover in each generation.

From the adoption of the Tabu search, inspired by the robotic foraging task: CAAM (Cellular Automata Ant Memory) by Lima and Oliveira, the authors Souza and Lima, in 2019, [75], proposed the TIACA, Tabu Inverted Ant Cellular Automata, which sought to detail the context of the use of a Tabu memory that locally assisted the repetition of paths already traversed by the robot. However, for these experiments, only 1 robot was used. To solve this problem, the authors Lopes and Lima, in the work called ETIACAM, Evolutionary Tabu Inverted Cellular Automata with Maps [76], added a map that helps robots to guide themselves through the environment, and in the same work, the authors investigated the model called GETIACAM (Greedy Evolutionary Tabu Inverted Cellular Automata with Maps) in which IACADI's greedy and elitist strategies were presented, what differs is again the addition of maps for each room to help the robot to move around, but there is one considerable expenditure of memory and computational time when adopting this strategy.

In this work, we created the GSTIACA model, Genetic Shared Tabu Inverted Ant Cellular Automata, in which we sought, in addition to adding the strategy of a memory based on the Tabu Search, we made a global sharing of this memory, which differs

¹ In our work we performed a part of the synchronous transition rule for robot movement and execution of movements and cell update and another asynchronous part, which was used for the pheromone decay update. This asynchronous update was used only because in our work, this was a computationally expensive task and we optimized by using asynchronous updating of the cellular automaton.

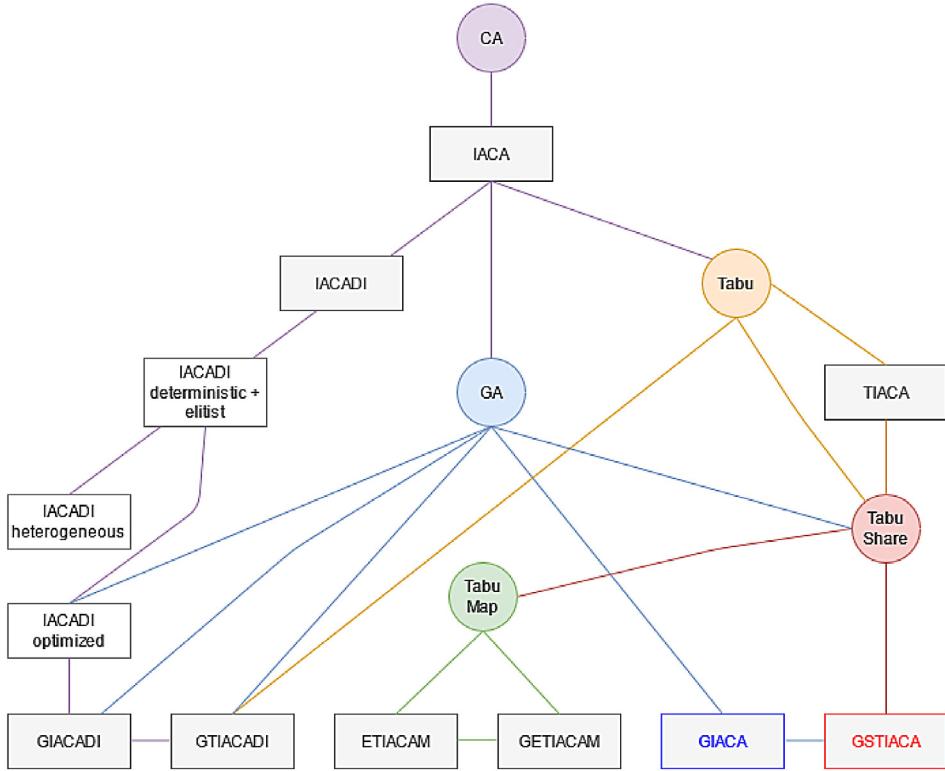


Fig. 1. Tree diagram representing the state of the art navigation models evolution for robot swarms based on cellular automata.

from all precursor works published to date. In this case, the robot has knowledge of all the places recently traveled by the other robots of the team within the same work environment (room), in a global way, in addition to the places recently traveled by itself, in a local way. That is, the strategy of looking at the queues of other robots within the same environment considerably helps the robot team's performance, since although the pheromone gives the robots the idea that some cells have been recently visited, it may be that they are still traversed by robots, since the selection of the next cell within the CA lattice is a random strategy. However, when we place robot cells within the same room as tabu, we reiterate that those cells within the same room should no longer be recently visited by the robot team. Thus, in this work, we were able to implement a strategy with good performance and good use of computational resources for the architecture of e-Puck² robots in the Webots EDU simulator. A detailed explanation of the model will be presented in the next section and simulations will be presented later.

2.2. Techniques that inspired the model

An ant colony can thrive for decades, changing its behavior from learnings that occurred in past events, even with the death of the ants [41]. The anthill can have very complex structures. Ants are social insects and depend on the division of labor to maintain their colonies. The workers perform various tasks, such as looking for food, monitoring the environment, cleaning the anthill, taking care of their ant larvae [41]. Ants communicate indirectly through their pheromone, detected with their antennae. From this communication, they are able to recognize trails that lead them to food sources or warn of possible predators, in this case, the pheromone works in a repulsive³ way [6]. Interactions with the environment and with other ants can be positive or negative.

In line with this same context, pedestrians also exhibit social behavior similar to ants when they are together. Pedestrians can exhibit panic or herd behavior when they are in a group, especially when people evacuate [36]. Through the behavior of these pedestrians, a model for robotics was created for the task of foraging that was inspired by this theme and also by the repulsive behavior of ants. This social behavior can be simulated by computers, through different techniques, and one of them is the CA [4]. CAs also have the nature of exhibiting an emergent dynamic behavior over time, as well as pedestrians, who exhibit behaviors close to exits, such as congestion [36], bottleneck [31], herd effect [30], arching [36], queuing [30], among others. The CA movement

² Publications that investigate e-Puck robot (chronological order) are listed in the following e-Puck's website: http://www.e-puck.org/index.php?option=com_content&view=article&id=34&Itemid=36

³ The pheromone is used as a mechanism to scatter robots, since the more pheromone present in an AC cell, the less likely the robot is to visit the cell again at any given instant of time. Additionally, the repulsive pheromone makes the robots scatter in the environment, increasing the hiding power and improving the efficiency of the model.

rules and conflict resolution of this work were based on the pedestrian evacuation works of [30,31]. A similar behavior can be seen with ants when they are in panic and all these forces of attraction, repulsion and friction can be modeled using CA, which is an excellent tool for modeling dynamic systems of a chaotic or complex nature, as in the end a behavior emerges that is capable of solving certain tasks [77–79]. Furthermore, they are highly parallelizable and simple to implement computationally.

The model proposed herein, GSTIACA, is based on the Inverted Ant Cellular Automata algorithm, or shortly IACA [69], and herein, IACA had its parameters optimized, becoming Genetic Inverted Ant Cellular Automata, or GIACA model. GIACA was used as inspiration for GSTIACA, and also the Tabu Inverted Ant Cellular Automata algorithm, or TIACA [75]. As far as we know, it is the first model that integrates and merges several bio-inspired models such as, (a) cellular automata with local search based on Tabu Search, (b) Global search based on Ant Colony Optimization and hybrid search based on (c) Genetic Algorithms, for the surrogate parameters optimization, and (d) memory sharing Tabu, mimicking ants' direct communication, in one single model for robots communication and navigation. Although CA-based models have been proposed previously for other tasks in robotics [4] or just to imitate the pedestrians behavior (as a simulation type) [30], in this work we combine several triumphant algorithms from natural computation in a single robotics control algorithm. One of the main surveillance task stages is the coverage of all rooms in the environment [44]. The coverage model here was inspired by the ants' pheromone repulsion, which is the indirect communication between agents, and represents the global search mechanism [49]. The main novelty herein is the direct communication that is made by exchanging messages between robots from a certain CA radius, which can also be considered a local and global search mechanism, at the same time, such as a PSO (Particle Swarm Optimization) strategy [5,80]. In addition, a shared short-term memory based on Tabu Search [47,48] was also used in the search process to prevent recently visited cells from being patrolled again by the current robot and other robots sharing the same room. This improvement returns a better robots' spread in the environment, transforming the coverage search in a global and local approaches. Finally, we will use SGA-based optimization so that the search for the best parameters is effectively achieved. The differential of our model in relation to all its precursors [49,69,75] is precisely the direct communication by queues usage, based on Tabu Search [47,48] shared between several robots. There are lot of ways of configuring the parameters optimization automatically, such as, the PSO [5] or Simulated Annealing [81], the first approach deals with a evolutionary computation that iteratively optimizes a problem by trying to improve the candidate solution with respect to a given quality measurement, the second approach is a meta-heuristic for optimization that consists of a probabilistic local search technique, and is based on an analogy with thermodynamics. But herein we used the GA meta-optimization technique of evolutionary computation [70], that is the main idea of our proposed paper, that uses bio-inspired methods in its construction. In addition, GA was also used in the construction of [74] work and in [73] work, which were also bio-inspired approaches. We use an evolutionary approach for comparison purposes with earlier works. In the aforementioned work [74] and in the work proposed herein, 10,000 CA iterations over 100 CA generations were used as a way of evaluating the 300 individuals GA with an average of 10 for fitness calculus, considering the mean and standard deviation, that is, $3,000,000,000 (3 \times 10^9)$ runs in total, which resulted in a run with an average of almost 24 h for each of the two models simulations, the GIACA and the GSTIACA. As a novelty in our computational model, we use a queue sharing mechanism between robots within the same environment, in addition to using evolutionary computation to optimize parameters. Besides that, it is possible to highlight the asynchronous cellular automata usage in pheromone decline process, improving the algorithm performance. We used a greedy randomized adaptive search procedure [82] for robots' CA cells choice, called here robot first choice movement [30]. We also used models to mimic the social pedestrians behavior during panic evacuation to model possible conflicts between robots [30,31,36,83]. Finally, we simulated our control model in Webots EDU using the e-Puck architecture [84] to validate that our model is capable of being implemented in a real architecture.

3. Proposed method

This section describes the surveillance task proposed model, using natural computing triumph strategies. As long as it was studied, no previous work has drawn an algorithm that merges different search techniques that include local, global and meta-optimization search methods, in order to use the natural behavior observed in the insects dynamics [40] and pedestrians [85] to guide the swarm robotics at the desired behavior [86]. At the same time make a natural selection via SGA [63] and only the best parameters are selected for the search be performed in an optimized manner. The integration of a range of bio-inspired algorithms [87] and intelligent strategies: cellular automata, Tabu search, greedy randomized adaptive search procedure [82], collective ants [40], and meta-optimization by surrogate genetic algorithms, using inverted pheromone and swarm robotics [4], was not found in the literature in an unified way in a single algorithm to solve the surveillance problem according to the robotics context, making the proposed model a hybrid and intelligent approach for cooperative scenarios. It is known that many artificial systems, including robotics approaches, are able to imitate behaviors that are successful in nature to cooperatively solve specific tasks. Following this same reasoning, the new model presents an integrated strategy that uses not only natural triumphant behavior, but combines many successful collective approaches to problem optimization. This combined social strategy fits in with robotics to solve surveillance, using multi-agents that interact cooperatively through local interactions, emerging global behaviors for patrol performance. In addition, it is known that the robotic surveillance task is comparable to several other tasks, such as covering environments, mining, foraging, and even space exploration.

3.1. System overview

The system will be explained below and has four defined stages: (1) individual behavior of each robot in the system, (2) global system guided by the cellular automaton through the structures sharing between all intelligent agents, (3) meta-optimization of

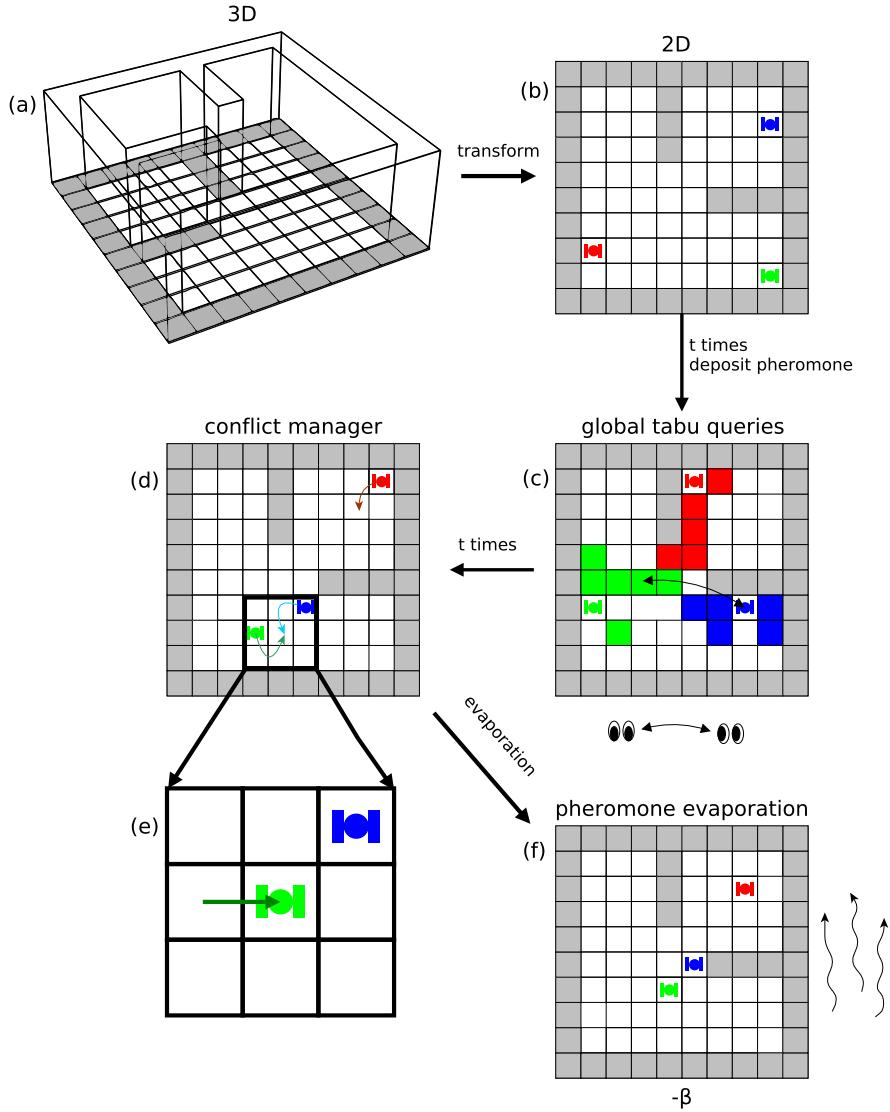


Fig. 2. Flowchart showing the two-dimensional (2D) cellular automata representation, queues sharing, conflict resolution and pheromone evaporation, where robot p_1 is represented by (■), robot p_2 is represented by (□) and robot p_3 is represented by (■) symbol. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

system parameters controlled by a genetic algorithm, and finally, (4) asynchronous cellular automata update for communication between robots, to optimize processing time.

Initially, a general 3D simulation environment is discretized into regular cells with the same size, representing the CA lattice (L) or grid. Subsequently, the rooms number in the environment is calculated, according to Fig. 2a, which in this case has 2 distinct rooms (Ω). Then, a spatial environment transformation is performed in such a way that it starts to be represented in a 2D two-dimensional projection, Fig. 2b. Some CA cells (x_{ij}) represent obstacles or walls and have their states defined as $x_{ij} = \infty$ and free cells with their states defined as 0. Each robot is positioned in a different cell with a different label value. During the CA time $0 \leq t \leq T$ we have a change in the pheromone, which initially is worth 0 and can be increased later. Then, a N amount of robots is used to carry out the surveillance, according to Fig. 2c. Each robot has in its data structure a row responsible for storing the steps covered by it (CA grid cells: row and column), each robot $\{p_1, p_2, p_3\}$ in the image has a colored cell that corresponds to the robot's color. Note that the robot in Fig. 2c is deadlocked, that is, it is in a corner (└) and stuck in its own Tabu memory Q . Then, according to Fig. 2d the robot chooses a cell surrounding its Moore neighborhood to move around its neighborhood η_{ij}^v , from a radius r , where $v = (2r + 1)^2$. Conflicts are solved, when necessary, Fig. 2e. In this case, if two robots want to use the same cell [30], where only one of them will perform the movement, the green robot wins the non-deterministic dispute, Fig. 2e. Finally, in Fig. 2f, the pheromone, which serves as indirect communication, is being decreased over time to improve the indirect agents communication.

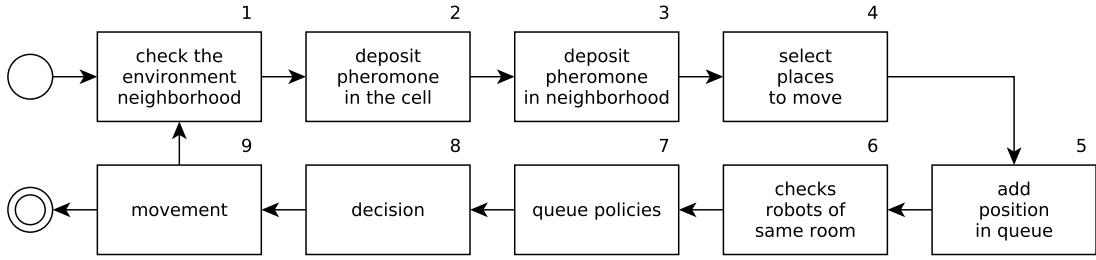


Fig. 3. State representation diagram of the robot's individual behavior.

3.2. Individual behavior

The robot's individual behavior can be understood as a finite state machine (FSM) and is shown in Fig. 3. This diagram describes each state through which the robot goes.

Fig. 3 defines the robot positioning, which is described below: initially, the robot is placed in an environment cell x_{ij} , this cell represents the CA grid. Subsequently, the robot deposits a pheromone in its central cell x_{ij} (2) with a δ value, which at a later time will be read by all other robots. Then, the robot, from the CA Moore neighborhood η_{ij}^v (where $v = (2r + 1)^2$), deposits an amount σ in its neighborhood (3) depending on the r value. This accumulated pheromone will be a kind of smartgrid that will serve for the global search system behavior. The robot tends to act doing a greedy adaptative decision [82] (or called as first choice decision [30]). From there, the robot defines the possible places to move to (4). The robot then adds the current position to its Q queue (5), that is, the current row and column (current position). This queue is an auxiliary memory that was inspired by the Tabu search [48]. This queue aims to assist in the robot's local search, in an attempt to improve the system as a whole, so that the robot does not return in recently visited positions.

Then, as a search optimization mechanism, the robot tends to look at the other robots queues $\forall Q_{p_i}$ (6), as long as $i \neq n$. This mechanism helps the robot not to browse places recently visited by other robots looking in its queues, improving local and global search, that is one novelty of our model. In order not to slow the simulation system, only robots within the same room are evaluated, that is, within a CA radius that does not exceed the boundary of the Ω room. This also allows the model to be implemented within a real context, where robots need direct communication protocols for data sharing (mimicking ants behavior [41]). In this case, these data are the positions recently traversed by other robots within a CA radius r , that represents the same room Ω , being that they exercise the function of improving the communication signal and also the queues sharing.

Thus, the robot has a restrictions Tabu queue Q (Tabu memory that includes its own recently visited positions), and the positions recently visited by other robots into its room, and the obstacles, these being the positions that cannot be part of the robots' choice.

The robots' queues behavior is defined following the queuing policies according to [68]. The queue policies are applied when the queues are either in a completely full state (when it reaches its maximum size), or in the deadlock state, which is when the entire robot's neighborhood is filled, blocking its movement, so that it cannot move until the queue state is changed. The states that activate queue policies are as follows:

QP_{∞} Deadlock case: when the robot is in deadlock state inside a corner \vdash , within its own queue Q or from sharing another robot's queue over a distance Ω

QP_{\emptyset} Full case: when a given robot's queue is full.

From the state that activates each queue policy, you can choose the empty queue option, which removes all items from the queue, or the dequeue option, which removes the last element in the queue. The queue policies, which will be further refined, defined by our GSTIACA model, which are in line with the optimization achieved by the genetic algorithm are: queuing in both cases, both full case and deadlock case [68]. The robots select the next CA lattice cell based in greedy randomized adaptive search procedure [82], where they can select one cell x_{ij} into its neighborhood η_{ij}^v . Robots are more likely to go greedily to less visited cells around their neighborhood. However, it is a randomized process in which they have a non-zero chance of choosing other cells around their neighborhood, as long as it is not a CA Tabu cell.

3.3. Global behavior

The system has a global state in which its structures are shared among all agents. In this case, the global behavior is achieved through the control system of the cellular automaton that emerges a global complex behavior. Additionally, we have the layers for local and global mechanism for indirect and direct communication via artificial pheromone ants. We have other layer to represent the robot cells states, such as free cell, cell occupied by a robot, obstacles cells. In addition, we have a partial sharing of the robots, based on the room. Initially, the real environment is divided into a discrete two-dimensional (2D) structure with cells identical to each other, as shown in Fig. 4a. A global CA clock is set to $t = 0$ when the simulation starts. This environment represents the CA lattice and has $x \times y$ cells. The theoretical size of each CA cell is considered to be 14×14 cm, which is twice the e-Puck architecture radius (≈ 7 cm) used in this work. Then, all cells are initialized with value 0 ($\forall x_{ij} = 0$) and $r = 1$ for all the simulations

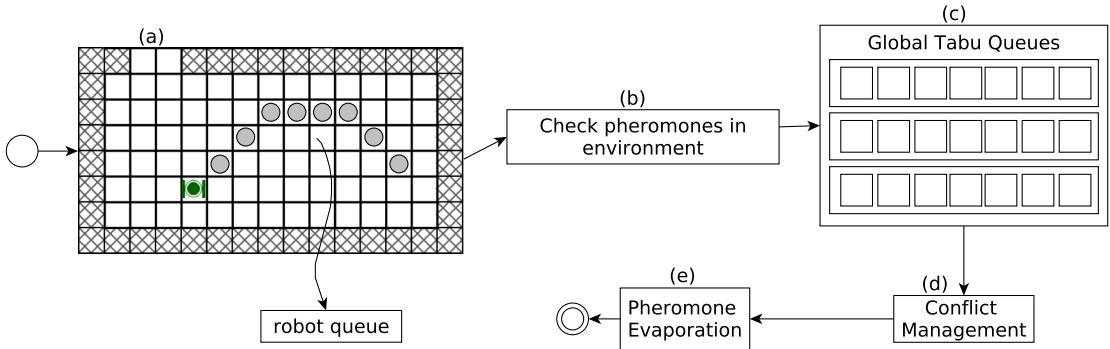


Fig. 4. Global CA behavior, demonstrating the order of pheromone deposition, queuing sharing, conflict resolution and pheromone evaporation.

and considerations in this article. As the robots pass through a certain cell, they check the pheromones that were deposited in the environment, as shown in Fig. 4b. However, each cell is the result of the contribution of each robot, according to Eq. (1).

$$x_{ij}^{t+1} = x_{ij}^t + (\tau_{max} - x_{ij}^t) \times (\delta \times e^{x_{ij}^t - \frac{\eta_{ij}^{v,t}}{\sigma r^2}}) \quad (1)$$

In Eq. (1) [88], v represents the robot neighborhood η_{ij}^v comprehends the range of $x_{(i+a)(j+b)}$, being $v = 2r + 1$ the robot radius $r = 1$ and (i, j) being the robot actual index that represents its position x_{ij} in the lattice L , with $(-r \leq a, b \leq +r)$. Also, σ is a constant for the pheromone propagation intensity, δ is the pheromone quantity deposited in the actual robot cell and τ_{max} is the total pheromone quantity that have to be accumulated in each cell L of the AC structure [69]. The parameter τ_{max} is a value that represents the addition of the pheromone in each of the CA cells, this value could be added up to a finitely large value. However, the value of τ_{max} brings changes to the system, weakening or increasing the global indirect communication time between the agents of the robot team. Therefore, this variable will serve as GA input, and will serve to optimize the navigation system as a whole.

Then, the robots must indirectly read the information from the floor field, always avoiding their recently visited cells, and sharing that information with robots in the same room Ω . At the same time, these robots must avoid paths recently visited by other robots. This means that this information is distributed among all the robots in the environment, according to Fig. 4c.

Conflicts must also be managed, which means that not always the intention of a given robot to move will cause it to make such a move, see Fig. 4d. This is because if two robots intend to move to the same cell, only one of them performs the movement. The algorithm for conflicts solver is based in [30,36,69]. Pheromones, which are a global system information, are used as indirect communication between robots, acting as an artificial chemical phenomenon, similar to the pheromone evaporation observed in biological systems, as shown in Fig. 4e. The Eq. (2) defines this evaporation factor in relation to the execution time. This evaporation serves to prevent robots from being trapped in local optima, being considered dynamic information, and providing guidance for the robots' actions.

$$x_{ij}^{t+1} = x_{ij}^t - (\beta \times x_{ij}^t) \quad (2)$$

3.4. Pheromones asynchronous update optimization

In this work we will use a global asynchronous CA update rule because the synchronous processing time is expensive to update the ants artificial pheromone used by robots as indirect communication. A set of tests were carried out on IACA and STIACA model to optimize using genetic algorithms (GA). GA need the simulations to be executed thousands of times in a short chronological time, so that they can quickly evaluate and optimize the solution. In the case of this specific surrogate model (the IACA and STIACA model) for GA fitness calculation, it was decided to make changes that did not affect the final solution, but that improved the execution time, eliminating unnecessary tasks.

Through the code analysis it was discovered that there was a high execution cost in the pheromones decay. The matrix that represents the pheromones consumes a large execution time due to the iteration throughout its length, each time the decay is recalculated. It was decided to recalculate only the robots' neighborhood, since only the neighborhood is used for decision making in the robots' movement. This change in the decay calculation guarantees a considerable improvement in the execution time, which improves the total time used by the GA, since it avoids the recalculation of parts that are less used by the robot.

The robot team control system makes global clock usage in which all robots use and access it. The clock is incremented with each CA iteration $t \in T$. Thus, as the robots team navigate the environment, it is important that it only update the cells that are in its neighborhood η_{ij}^v , from a certain CA radius r . Fig. 5 shows the neighborhood of each robot, where the time is correctly updated and where the remain of the matrix is not updated, also called lazy evaluation.

In order to update the pheromone decay for several time steps t , a small change was made to the decay equation presented in [69]. The original equation was created to update only one time step at a time, however, the change made allows updating several time steps at once in the pheromone decay. This causes all the code iterations that would occur for several steps for each

							00	04	05	06	07
00	00	00	00		01	03	10	10	10	08	
00	00	00	00	01	02	03	11	11	11	09	
00	00	00	00	01	02	13	13	13	12	09	
00	00	00	00		02	13	00	13	12	09	
12	13	13	13	00	00	00	00	00		00	
12	13	00	13	00	00	00	00	00		00	
11	13	13	13	00	00	00	00	00		00	

Fig. 5. Time measurement updated in each lattice cell. The regions in the robots' vicinity have their time updated, while the other regions, whose information is not used, remain without the pheromone level being recalculated.

cell at each time point to be replaced by a single calculation which is performed only in the neighborhood cells, as shown in Eq. (3).

$$x_{ij}^{t+m} = x_{ij}^t \times (1 - \beta)^m \quad (3)$$

Eq. (3) describes the advancement of pheromone decay in a single cell by $t + m$ time steps, m being the steps number that the cell remained without updating and t the current time. Assuming the temporal state of a cell is $t = 18$, and this cell remained for $m = 7$ time steps without updating, the current time is $18 + 7 = 25$, so $25 - 18 = 7$ pheromone decay time steps will be calculated, so the cell is updated to the current time. The Eq. (3) was built based on Eq. (2), originally used in the IACA model [69].

The described modification demonstrates that it is possible to achieve significant improvements in the execution speed through simple modifications in the algorithm. These modifications are necessary to improve the GA final execution. Assuming a matrix $L = 20 \times 30$ being used as a simulation environment (CA lattice), where 600 iterations would be performed per time step, when we do 10^4 iterations this number increases to 6.0×10^6 , regardless of the robots number. The modification made considering the neighborhood of 3 robots performs only 24 iterations per time step, for a total of 2.4×10^5 . This means that the larger the simulation environment, the greater is the repetitions number that are necessary for the CA lattice to be correctly calculated. In this specific case, with the approach programmed on our work, we complete the task 25 times faster. If this change had not been implemented, the CA parameters optimization of the navigation model by metaheuristics using GA (which can have a relatively slow execution) would not be feasible.

3.5. Refining the parameters using the genetic algorithm

This work uses parameters optimization by a genetic algorithm, one of the most used techniques in evolutionary computing. The GA provides the optimization of the parameters of the surrogate IACA, proposed in [69], and STIACA, proposed herein. After this meta-optimization, we started to call IACA as GIACA, and STIACA as GSTIACA. GSTIACA is the main model addressed in this work, but both can serve as a surrogate for an implementation with real architectures. This means that the GA is performed as a model pre-processing task, so as not to increase the simulations computational time. Then, we use the optimized parameters in the GA as input for the GIACA and GSTIACA models, and these parameters are no longer changed for the same simulation environment. After that improvement, GIACA is a better approach than IACA, and GSTIACA is better than both.

Each population size is equal $T_p = 300$ and has a set of individuals $d[v_1, \dots, v_m]$, where m is the parameters size, ($m = 4$) in GIACA and ($m = 8$) in GSTIACA. Each individual is tested $k = 10$ times, with different seeds for initialization. These tests are necessary because the surrogates (IACA and STIACA) on which we are based, are non-deterministic methods in which the measurement for a single time can guide us to the wrong choice of the best individual. The GA individual's goal is to maximize the completed tasks number denoted here CT_i , with $1 \leq i \leq 10$. Considering a CT is the number of completed tasks by robots, that means, at least one robot patrolled each of the rooms. When all rooms are patrolled at least once by at least one robot, it means that the cycle ends and the CT value can be increased by +1 value. The individual's general aptitude is given by the arithmetic mean of $k = 10$ simulations. The value 10 was chosen because it is the smallest value measurable from the standard deviation [73,74]. Eq. (4) shows the individual fitness (F_d) from the GA, d represents each descendant generated by crossover, and the general fitness is given by the arithmetic mean of the 10 internal simulations by the surrogates (IACA and STIACA). CT_i representing the tasks number completed

Table 1
Input CA parameters that is also GA individuals.

Number	Name	Lowest value	Highest value	Type	Description
1	σ	0.0	1.0	real	pheromone rate into current cell
2	δ	0.0	1.0	real	pheromone rate around current cell
3	τ_{max}	10.0	50.0	real	max quantity of pheromone for cell
4	β	0.0	1.0	real	pheromone evaporation rate
5	\overline{CQ}	0	1	boolean	check other tabu queue
6	QP_{\otimes}	0	1	boolean	tabu queue policy deadlock case
7	QP_{\emptyset}	0	1	boolean	tabu queue policy full case
8	$ Q $	0	50	real	tabu queue size

within $T = 10^4$ steps, which was the total steps number T of the CA-based surrogate model (IACA or STIACA) chosen based on [49].

$$F_d = \frac{\sum_{j=1}^k \sum_{k=1}^T CT_j}{k} \quad (4)$$

For our navigation model, each of the $m = 8$ parameters $\{\sigma, \delta, \tau_{max}, \beta, \overline{CQ}, QP_{\otimes}, QP_{\emptyset}, |Q|\}$ represent the parameters of an individual. For the GIACA model we have an individual with $m = 4$ genes $(\sigma, \delta, \tau_{max}, \beta)$. For GSTIACA we have a GA individual with 8 genes, being $(\sigma, \delta, \tau_{max}, \beta, \overline{CQ}, QP_{\otimes}, QP_{\emptyset}, |Q|)$. These parameters, which represent the genes, are first initialized with a default value that can vary between a minimum and maximum value, and such variation assumes different configurations depending on the parameter to be optimized. These values are defined according to Table 1. For the parameter display in the graphic that shows the GA parameters convergence, the values $d_n[v_1 \dots v_m]$ are individuals d genes normalized between [0,1] [89] using Eq. (5).

$$v_{norm} = \frac{v_i - \min(v_1 \dots v_m)}{\max(v_1 \dots v_m) - \min(v_1 \dots v_m)} \quad (5)$$

The value described as v_i is a gene of individual $d_i[v_1, \dots, v_m]$, with $(1 \leq i \leq n)$, is the value to be normalized among the possible $d_n \in T_p$ individuals values, with n being the T_p population size. This value is then decreased by the lowest value in the sample $\in T_p$. It is noticed that each sample value can vary according to the GA parameters, as can be seen according to the Table 1. Then the values amplitude is calculated by finding the difference between the highest value in the sample $\max(v_1 \dots v_n)$ with the lowest value in the interval $\min(v_1 \dots v_n)$.

Subsequently, each of the controller system parameters are guided by the CA and given as input to the genetic algorithm. Also, for the individuals choice in the population by the genetic algorithm, the same normalization is done before defining the probability of selecting an individual.

The genetic algorithm also has some parameters $\{P_{cross}, P_{mut}$, crossover type, selection type, reinsertion type}, etc. In this work, they were configured (see Fig. 6) in such a way as to contemplate the studies done in [55,58,74,90]. Fig. 6 shows the GA optimization flowchart and this model will be optimized by the GA.

As previously presented, the GA individual was made using a vector of 8 parameters (8 genes), and the parameters types, and their minimum and maximum values, which can be initially configured in the individual, are specified in Table 1. In this case, it should not be confused with the minimum and maximum values achieved by each generation as indicated in Eq. (5). The values in Table 1 serve as the minimum and maximum intervals through which the GA can initialize its parameters, as well as the values by which the IACA and STIACA surrogate models can operate. GIACA considers in its implementation the values in the Table 1 column "Number", between 1 and 4, considering only IACA [69] parameters. On the other hand, the main model proposed in this work (GSTIACA) considers the values from 1 to 8, in addition to the basic IACA parameters, the queue $|Q|$ parameter was also considered, as shown [75]. We brought as a novelty 4 new parameters, the \overline{CQ} , which evaluates the other robots queues within the same room Ω , the value QP_{\otimes} that evaluates the queue policy used in deadlock case, QP_{\emptyset} which evaluates the queue policy employed when full case occurs and $|Q|$ which defines the tabu queue size [68].

The variables σ, δ, β are represented by real values between [0,1]. The values $\overline{CQ}, QP_{\otimes}, QP_{\emptyset}$ are represented by Boolean values. On the other hand, the value τ_{max} , varies between the intervals [10,50], which is the maximum value that can be increased per pheromone cell. The variable $|Q|$ can vary between [0,50], which means that we can have an individual queue of size 1 ranging up to 50, or simply not have a queue configured when the value is 0. Thus, the individual for the genetic algorithm is represented by real numeric types, and, therefore, operations on individuals (for example: recombination and mutation) will be performed by methods that include individuals of real numeric types.

In this case, the genetic algorithm uses the global selection by roulette, followed by linear recombination and elitist reinsertion. According to [58], elitist reinsertion (which is a subset of global reinsertion) means that the offspring produced has a smaller population than the parents number selected for crossover and that the worst parents will be replaced by their children (without doing fitness comparison (F_d) between parents and children). In addition, normalized fitness was used, which influences the selection by roulette, making it more uniform, that is, distributing the odds equally among the entire population, which allows for greater characteristics variability. Normalization was done according to Eq. (5) [89], improving the fitness assessment.

Selection: for the parents' selection, an ordered pair (d_1, d_2) , the best individuals (greatest aptitude) are selected to generate offspring d' through crossover and mutation. In this work, we use the parents selection based on the individual's fitness.

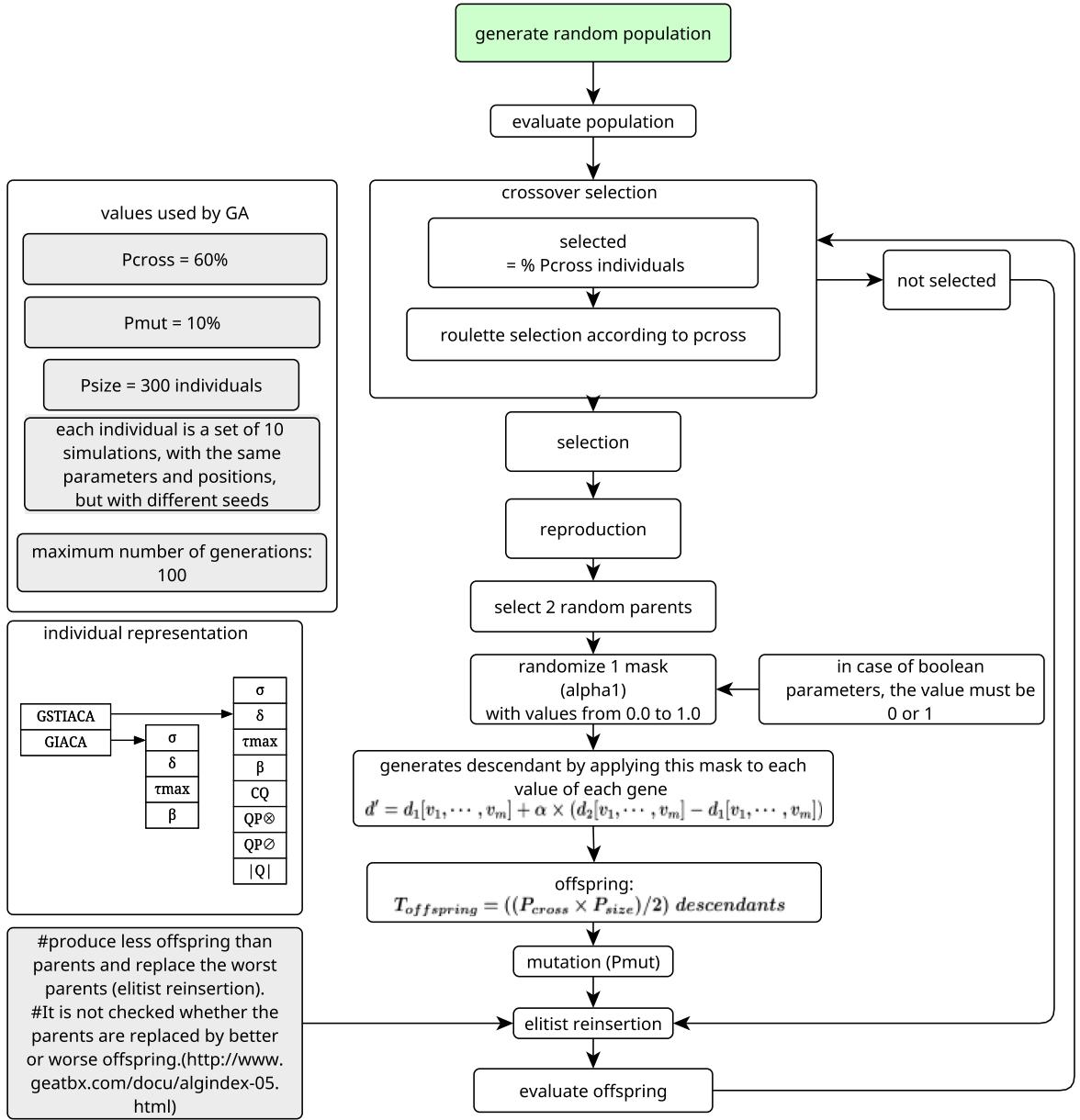


Fig. 6. Flowchart showing the operation of the genetic algorithm implemented as an optimization model. GA configuration variables are also presented, as well as stopping conditions.

For the parents selection, roulette selection is used, which is done as follows: a parents number is calculated to be selected according to T_{cross} , this amount being equal to $P_{cross} \times T_p$ [58], in addition, the fitness of each individual is normalized between the values [0.1], to allow a better distribution of the selection by roulette.

Crossover: the crossover combines selected individuals (parents), an ordered pair (d_1, d_2) , to produce children d' and allows you to explore unknown areas of the search space. In this work, we use crossover for real individuals. Therefore, the [58] linear recombination method was chosen, which, in this case, will generate one child for every two parents [91], resulting in an amount $T_{offspring} = \frac{T_p \times P_{cross}}{2}$ of children generated. If there is no mutation, the individual fitness F_d is calculated.

Mutation: the mutation applied to the GA guarantees a local search, that is, in this case, we use the alteration of a chromosome gene, drawing an increase/decrease of up to 30% from the variable value, in the case of real values, and a boolean value where can be {1 or 0} for boolean parameters. From the children generated d' in the crossover step, a mutation is applied to the $T_{offspring} \times P_{mut}$ children number [58], the individual fitness F_d is calculated.

Reinsertion: the reinsertion step consists in the new individuals generated (offspring) in the population that was not selected for the crossover. Elitist reinsertion is used, which consists of producing an offspring less than the parents number, and the worst parents are replaced by descendants without making a comparison between the parents and descendants fitness. From there, the individuals resulting from this process are joined with the population that had not been selected for crossover [58].

Fig. 6 details the GA operations. The algorithm starts by randomizing a population of individuals. From there, the first population assessment is made. After that first evaluation, a percentage P_{cross} of individuals is selected, through selection by roulette. Then these individuals go through the reproduction stage, which consists, first, in selecting two random parents d_1 and d_2 and randomizing a mask (α) with values between [0,1], as described in the GA with real individuals [58]. Thereafter the chromosome values of the descendant d' are defined according to Equation $d' = d_1[v_1, \dots, v_m] + \alpha \times (d_2[v_1, \dots, v_m] - d_1[v_1, \dots, v_m])$, where $d_1[v_1, \dots, v_m]$ is the parent gene value d_1 , and $d_2[v_1, \dots, v_m]$ is the parent gene value d_2 . Thus, a descendant d' is always generated for every two parents, to facilitate the children evaluation. The resulting individuals number of the next generation (offspring) can be calculated by the equation $T_{offspring} = \frac{P_{cross} \times P_{size}}{2}$. Then, as can be seen in the flowchart of **Fig. 6**, a random mutation is inserted in the descendants (from a mutation rate T_{mut}), and the descendants are reinserted back in the population. Finally, only individuals who have been reinserted need to be re-evaluated, and the algorithm is repeated until reaching the stopping criterion. This entire process is detailed in **Fig. 6** and the algorithms presented below complement the figure to explain the GIACA and GSTIACA models.

The main idea of the GIACA and GSTIACA models is demonstrated in the Algorithms 1 and 2, respectively. For the GIACA demonstrated in the Algorithm 1 we have that the algorithm's return comprises the best set of genes that represent the parameters to be used later in all other simulations for model refinement and verification of flexibility, efficiency, robustness and effectiveness of it. Initially, it is possible to see that the algorithm consists of 16 steps, with the steps consisting of initializations, repetition structures and conditionals. Initially, the parameters of the GA are initialized and the first looping structure of GIACA consists of looping representing the $G = 100$ generations of the GA. Thereafter, all d individuals are evaluated and each individual's fitness F_d is calculated.

The fitness calculation in GIACA is performed over $T = 10^4$ iterations in order to calculate the average of 10 different initializations of the completed tasks CT number. As a reminder, a completed task CT is equivalent to the sum of task points TP , whose objective is to validate when all rooms have been visited by at least 1 robot, thus ending 1 complete patrol cycle. Furthermore, the pheromone deposits are properly calculated, as well as the lazy evaluation for the pheromone decrease. With the use of lazy evaluation, the order of complexity substantially reduces, that is, instead of using at each iteration t an update in $O(L)$, where L is the lattice size, in this new approach the calculus becomes of $O(N)$, where N is the maximum number of robots, and $N < L$. The pheromone deposit, on the other hand, considers that the robot places δ in the central cell x_{ij} costing $O(1)$, and δ' in its vicinity η_{ij} costing $O(8)$, with a maximum size $O(9)$, which, since it is constant, and will be discarded from the calculation for asymptotic notation. The robots are moved at each iteration t with the appropriate probability calculations performed properly so that there are no conflicts between the robots, with the cost of computational time $O(T)$. For each iteration t , an average of 10 runs is performed to calculate the fitness F_d , this average for the fitness calculation has a computational cost of $O(10)$, which is constant. Therefore, it will be omitted in our asymptotic complexity calculation. At each new generation i a set of all individuals $d \in T_p$ are generated from the crossover and selection of parents (d_1, d_2), which are reinserted and properly ordered with a computational cost of $O(T_p \log T_p)$, using an optimal sorting algorithm. With these generations, the refinement and evolution of genes that will serve as the algorithm output and that will be adopted later in order to refine the GIACA navigation model, whose computational time cost is $O(G \times T \times T_p \log T_p)$, considering the main costly variables for the algorithm.

The main difference between GIACA and GSTIACA presented in the Algorithm 2 is precisely the addition of the Q queue, whose objective is to increase the model efficiency in relation to the number of patrols, since the queue is a parameter that prevents a robot from avoiding places recently visited by itself and by other robots within the same room. The total number of steps of GSTIACA is 20 steps, as shown in the Algorithm 2. The computational cost to perform this task depends on the number of robots N and the maximum queue size $|Q|$. Since the GSTIACA algorithm consists of scanning the queue of all robots at a given time t , depending on the memory size and the number of robots, with computational cost of $O(N \times |Q|)$. Hence, considering previous explanations and discarding some fixed or constant value operations, the GSTIACA algorithm has a computational cost of $O(G \times T \times T_p \log T_p \times N \times |Q|)$. The addition of the queue helps in the robot team performance and environment coverage, as it supplements the flaws left when using only the lattice with the pheromone. In the next sections the models evaluation experimental results will be presented and discussed.

4. Experimental results

In this section, the main experiments for validation and model refinement will be presented, among which we can mention: optimization by GA, statistical analysis results and environmental coverage (completed tasks and pheromones). This first part was carried out using the Python programming language. Finally, the experiments carried out on the Webots with the e-Puck robotic architecture, with a controller created for each robot in the C programming language. Each set of characteristics, which generally involve the way in which the decision is made, is tested and optimized, and evaluations of the optimized models are obtained. The first experiments served to refine the model parameters and the experiments with robots indicate that the algorithm can be implemented in a real architecture.

Algorithm 1: GIACA algorithm.

Result: CA parameters represented by the set of 4 genes of the best individual d of GA

```

int  $i = 0$ ;
(1) initialization of all GA parameters;
int  $d \in T_p = [1, \dots, 300]$ ;
int  $G = 10^2$ ;
while  $i \leq G$  do
    int  $T = 10^4$ ;
    if  $i \neq 0$  then
        (2) select parents  $(d_1, d_2) \in [1, \dots, 300]$  by roulette;
        (3) create individual  $d'$  by crossover  $(d_1, d_2)$ ;
        (4) apply a mutation in  $d'$  with 10% of probability;
    end
    int  $k = 0$ ;
    while  $k < 10$  do
        int  $CT = 0$ ;
        (5) initialization of all CA parameters according to  $d'$  or  $d$  genes;
        int  $p_{ij} \in [1 \dots N]$ ;
        int  $t = 0$ ;
        while  $t < T$  do
            (6) read central  $x_{ij}^t$  and neighborhood cells  $\eta_{ij}^t$ ;
            (7) calculate pheromone deposition  $\delta$ ;
            (8) deposit pheromone  $\delta < \tau_{max}$ ;
            foreach robot's  $p_{ij}$  cell  $x_{ij}^t \in \eta_{ij}^t$  do
                (9) calculate robot's  $p_{ij}$  movement probability using an inverted roulette;
                (10) probability is the inverted weighted probability of the  $\eta_{ij}^t$  cells;
            end
            (11) solve collision conflicts;
            (12) move robot  $p_{ij}$ ;
            if  $TP$  is completed then
                |  $CT = CT + 1$ ;
            end
            (13) global pheromone decay;
             $t = t + 1$ ;
        end
         $k = k + 1$ ;
    end
    (14) calculate individual  $d'$  or  $d$  fitness  $F_d$ ;
     $F_d = \frac{\sum_{k=0}^{10} CT}{10}$ ;
    (15) sort individuals  $d$  selected for crossover by fitness  $F_d$ ;
    (16) eliminate weakest crossover individuals  $d$  and replace them with offspring  $d'$ ;
     $i = i + 1$ ;
end

```

4.1. Configuration and operation of the simulation environment parameters

The first experiment carried out aims to refine the controller parameters modeled by the CA. The improvement performed by the GA in the final result consists in optimizing the control parameters algorithm of a robotic surveillance task. Several modifications are made to these algorithms, including the hybridization of their strategies, that is, several characteristics of different models have been combined into a single final model for robot navigation. In this context, GA finds the best parameters to be used, in the robot control and navigation algorithm called GSTIACA and for comparison we will use GIACA. The initial GA parameters were $\{T_p = 300, P_{cross} = 0.6, P_{mut} = 0.1, G = 100\}$ [55,58], as shown in Fig. 6. Both GIACA and GSTIACA were proposed by us here. However, GIACA is an IACA optimization parameters proposed in [69]. While TIACA [75] presents a modification in the IACA controller resulting in a model with the queues implementation. In this work, from IACA and TIACA we created a new controller called STIACA that consists of sharing Tabu queues, and this in turn, when its parameters are optimized by a GA, is called by us GSTIACA.

In this case, we will use, among the criteria for the analysis, the average of 10 simulations, to represent the fitness of each GA individual, as explained previously. The average (\bar{x}) is used to represent the GA fitness, and the convergence is represented by

Algorithm 2: GSTIACA algorithm.

Result: CA parameters represented by the set of 8 genes of the best individual d of GA

```

int  $i = 0$ ;
(1) initialization of all GA parameters;
int  $d \in T_p[1, \dots, 300]$ ;
int  $G = 10^2$ ;
while  $i \leq G$  do
    int  $T = 10^4$ ;
    if  $i \neq 0$  then
        (2) select parents  $(d_1, d_2) \in [1, \dots, 300]$  by roulette;
        (3) create individual  $d'$  by crossover  $(d_1, d_2)$ ;
        (4) apply a mutation in  $d'$  with 10% of probability;
    end
    int  $k = 0$ ;
    while  $k < 10$  do
        int  $CT = 0$ ;
        (5) initialization of all CA parameters according to  $d'$  or  $d$  genes;
        int  $p_{ij} \in [1 \dots N]$ ;
        int  $t = 0$ ;
        while  $t < T$  do
            (6) read central  $x_{ij}^t$  and neighborhood cells  $\eta_{ij}^t$ ;
            (7) calculate pheromone deposition  $\delta$ ;
            (8) deposit pheromone  $\delta < \tau_{max}$ ;
            foreach robot's  $p_{ij}$  cell  $x_{ij}^t \in \eta_{ij}^t$  do
                (9) calculate robot's  $p_{ij}$  movement probability using an inverted roulette;
                (10) movement probability is the inverted weighted probability of the  $\eta_{ij}^t$  cells;
                (11) movement probability is 0 for cells  $\in Q$  tabu queue;
                (12) movement probability is 0 for cells  $\in Q$  of other robots in the same room  $\overline{CQ}$ ;
                (13) insert CA parameters  $d'$  gene restrictions for full  $QP_\emptyset$  & deadlock  $QP_\otimes$  cases;
            end
            (14) solve collision conflicts;
            (15) move robot  $p_{ij}$ ;
            (16) add previous robot  $p_{ij}$  position in tabu queue  $Q$ ;
            if  $TP$  is completed then
                |  $CT = CT + 1$ ;
            end
            (17) global pheromone decay  $\beta$  considering lazy evaluation;
             $t = t + 1$ ;
        end
         $k = k + 1$ ;
    end
    (18) calculate individual  $d'$  or  $d$  fitness;
     $F_{d'} = \frac{\sum_{k=0}^{10} CT}{10}$ ;
    (19) sort individuals  $d$  selected for crossover by fitness  $F_d$ ;
    (20) eliminate weakest crossover individuals  $d$  and replace them with offspring  $d'$ ;
     $i = i + 1$ ;
end

```

population fitness [92]. In the model proposed herein, the algorithm stop criterion is the execution of a maximum of $G = 100$ GA generations, each generation with a population of 300 individuals. In addition, the fitness of each individual in the GA is the average of 10 CA environment simulations. Each of the GA parameters can be seen in the Table 1 and in Fig. 6 in the upper left corner. An amount of $N = 3$ robots was randomly initialized throughout the environment. The environment E_1 is the CA lattice with 20×30 cells (280×420 cm), as shown in Fig. 7. The environment E_1 was used to refine the parameters of our two surrogate models. Different forms of obstacles were placed in order to simulate the robotic surveillance task with greater veracity and reliability, thus generating an environment with (□) empty cells that represents 7 rooms, (■) 7 doors and (■) many cells that represents the obstacles or walls.

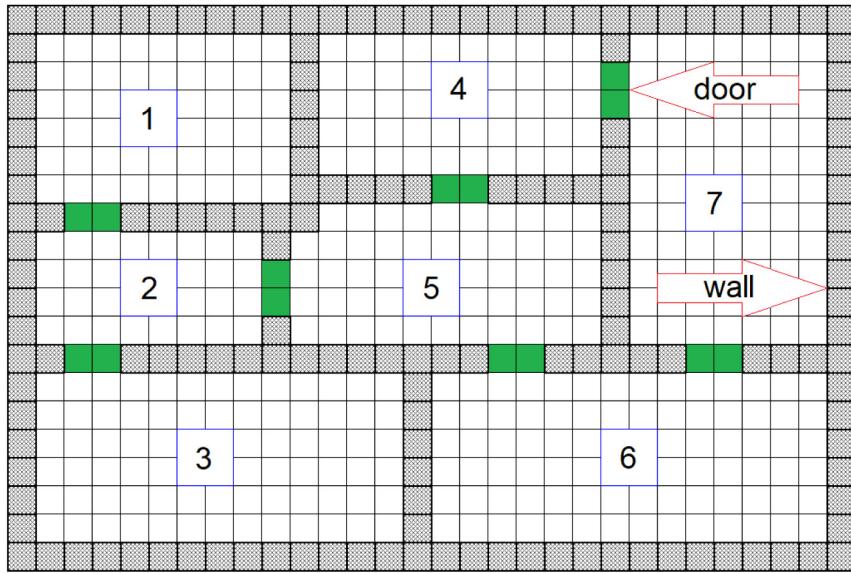


Fig. 7. E_1 environment used for the model parameters refinement using the evolutionary computation for the GIACA and GSTIACA models, with seven doors and seven rooms. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

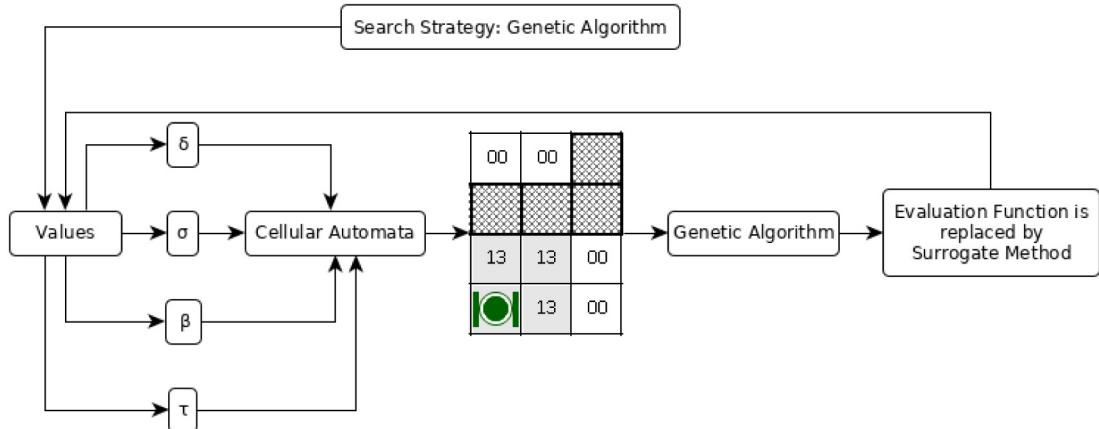


Fig. 8. Flowchart describing the interaction between Genetic Algorithm and Surrogate Method.

Fig. 8 exemplifies the interaction between cellular automaton (surrogate) and genetic algorithm for these experiments, demonstrating the parameter refinement process performed. **Fig. 8** shows the parameters ($\sigma, \delta, \beta, \tau_{max}$) data as input for the surrogate (which can be either IACA or STIACA) also become input GA measures, which in turn evolves these parameters to more optimized values. These values in turn will serve as input to the surrogate (IACA or STIACA) in a feedback mechanism. An example of this change in genetic algorithms is described in [93], which defines a modification in order to replace the evaluation function (fitness) with a surrogate simulations. The experiments proposal that will be detailed later is similar and uses, in this case, a CA simulation, as described in **Fig. 8**.

Fig. 8 shows all the GA functioning is maintained, including its hyperparameters, or input parameters. In this case, a flow chart showing part of the GIACA functioning with its respective parameters is being represented. After the GA initialization, the evaluation function is replaced by a surrogate method. In the specified experiment, the part that has the surrogate role is composed of a cellular automaton. This cellular automaton is used for the simulation of a surveillance task using robots, in an inverted pheromone system, as described in [69]. The cellular automaton implemented in accordance with [69] has several configuration parameters that can be changed according to the genetic algorithm. These parameters will go through the optimization process using a GA, aiming to reach a result close to optimum in the final configuration.

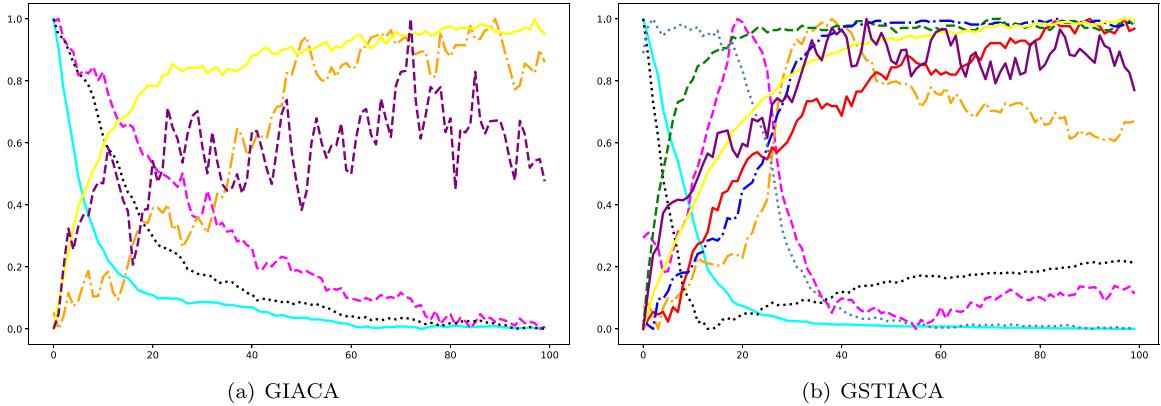


Fig. 9. The parameters are: (■) representing $\sigma \in [0, 1]$, (■) representing δ , (■) representing τ_{max} , (■) representing β , (■) representing check other tabu \overline{CQ} , (■) representing tabu policy deadlock case QP_{\otimes} , (■) representing tabu policy full case QP_{\emptyset} , (■) representing tabu size $|Q|$, (■) representing mean \bar{x} representing all individuals in population, (■) representing standard S deviation from all individuals in population. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Optimization of surrogate parameters using evolutionary computation

For each model, the parameters were refined from a GA and are represented in the graphs of Fig. 9. In each graphic it is also possible to see the GA parameters convergence, (y-axes) after 100 generations. The refined GIACA of 4 parameters are $\sigma, \delta, \tau_{max}$ and β . For GSTIACA the refined of 8 parameters are $\sigma, \delta, \tau_{max}, \beta, \overline{CQ}, QP_{\otimes}, QP_{\emptyset}$ and $|Q|$. Recalling that the minimum and maximum values were presented in Table 1. In addition to the parameters, the tasks average \bar{x} completed by the entire population and the standard deviation S of the GA population over the 10^2 generations are shown. A comparison is made between the GIACA and GSTIACA models. GSTIACA model has the global pheromone map (deposition and decline) usage for decision making [69], Tabu queue and queue policies were also used here, with the other robots queues verification, based on [4,66,68,75].

The GA parameter convergence diagram, Fig. 9, demonstrates mean values population parameters oscillation, as the evolution takes place. Each parameter displayed has its value normalized to a value between 0 and 1, as shown in Eq. (5), and each value is also the average result of that same value considering the entire population of 300 individuals per generation. Fig. 9 shows the direction in which the parameters converge, considering the entire population. Fig. 9(a) shows the GIACA convergence graphic with its 4 parameters, mean and standard deviation for all individuals in the population. While Fig. 9(b) we have the GSTIACA convergence model with its 8 parameters, mean and standard deviation for all individuals in the population. It is noticed that, after a few generations, these parameters tend to stabilize, indicating the convergence of their solution, that is, average \bar{x} stabilization, demonstrating that the GA reached a solution that does not change drastically in the population in general over the generations.

As a result of this experiment, we obtained the final values for GIACA and GSTIACA for each parameter, without normalization. These values without normalization will serve as input for surrogates IACA and STIACA. The final values found in GIACA without normalization are: $\sigma = 0.065$, $\delta = 0.163$, $\tau_{max} = 38.162$ and $\beta = 0.061$. In this case, it is possible to notice that as the GA evolves the values of τ_{max} tend to increase. On the other hand, the σ, δ and β values tend to remain smaller. For GSTIACA the refined parameters were $\sigma = 0.016$, $\delta = 0.859$, $\tau_{max} = 30.364$, $\beta = 0.322$, $\overline{CQ} = 1$, $QP_{\otimes} = 1$, $QP_{\emptyset} = 1$ and $|Q| = 14$.

The deposition values in the central cell δ converged to higher values than in the cells around the robot's neighborhood (from the σ value). On the other hand, for both models, the β decreasing pheromone value is also a low value (in this case, less than the central cell value) as well as the τ_{max} value. The queue policies adopted were a $|Q| = 14$ with the queue sharing usage enabled for robots within the same room. Additionally, every time the queue fills or the robot is deadlocked, an item in the queue is dequeued. Thus, these values will be used as parameters for the experiments in the next sections, as they were the optimal values generated by the GA.

4.3. Comparison performance between GIACA and GSTIACA models

The purpose of this section is to compare GIACA and GSTIACA models. First, a comparison will be presented through statistical analysis, using boxplots. Subsequently, a regression analysis will be carried out. Finally, two graphs of Task Points will be presented to evaluate the models. The environments in Fig. 10 were used to compare the GIACA and GSTIACA models are $\{E_1, E_2, E_3, E_4\}$, all with 7 rooms and 20×30 cells. Additionally, the values found after the meta-optimization will be used.

4.3.1. Statistical analysis

In this experiment, we aim to create boxplots – a diagram of descriptive statistics that is one of the main graphic tools to represent the observed data variation from a numerical variable – to check the average number of visits from the variation of rooms and robots. In this case the times number CT that the robots team N (x-axis) completes the surveillance task in a certain time T , by quartiles

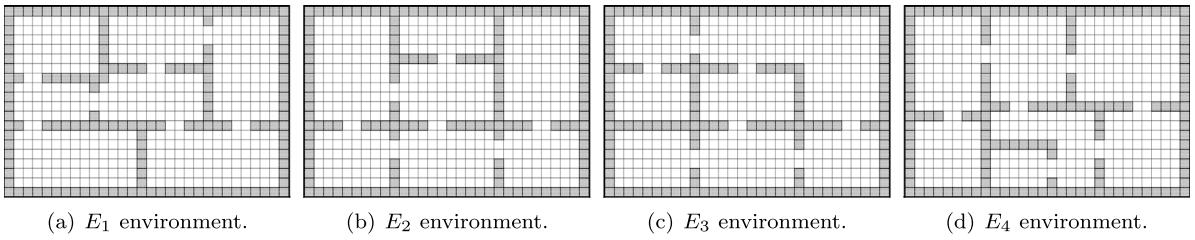


Fig. 10. Environments (CA lattices) for robotic simulations, all with 7 rooms using 20×30 cells.

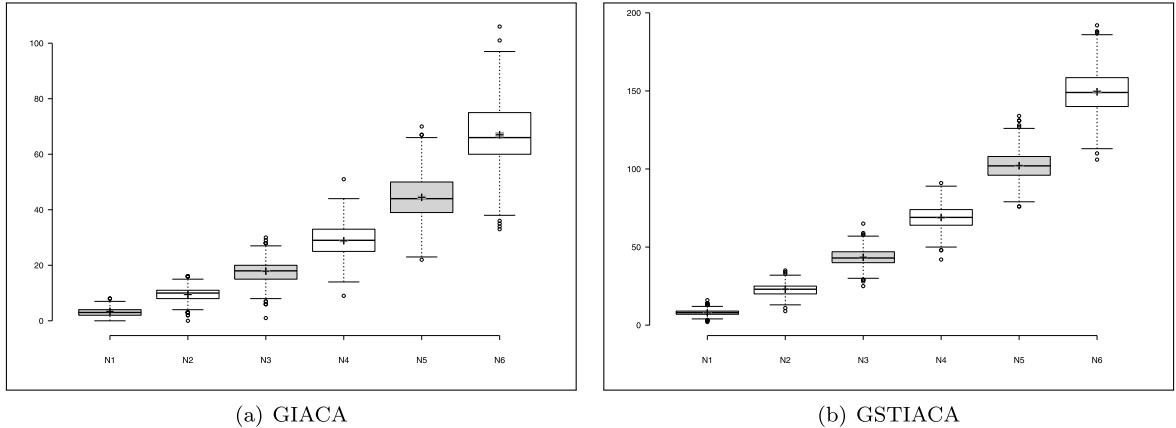


Fig. 11. Boxplots showing average visits for each $N = \{1, 2, 3, 4, 5, 6\}$ robots. E_1 environment.

means (y-axis), represented by the symbol (\square) that represents 50% of the simulated values. Outliers are represented by the graphic symbol (\circ) and upper whiskers (\top) and lower whiskers (\downarrow), which represent the highest and lowest values seen during the simulations disregarding the outliers. Figs. 11, 12, 13 and 14 show results comparisons achieved by GIACA and GSTIACA for each environment. When comparing these results, it is possible to notice differences in the tasks number accomplished as the environments change characteristics. For the results analysis, we use the average, represented by the graphic symbol (+), and it has a confidence interval of 95%, represented by the graphic symbol (\blacksquare). A total of 10^3 simulations were performed for each environment with $T = 10^4$ time steps each. This amount of simulations allows for better precision in the models measurements, when compared to the GA, which uses a minimum simulations number to evaluate the results. $N = \{1, 2, 3, 4, 5, 6\}$ robots were used, for a total of 7 rooms each and using the four environments E_1 , E_2 , E_3 and E_4 , as can be seen in Fig. 10 and Table 2, for more detailed results of the experiments.

In this case, the objective is to test characteristics of each model and finally use those characteristics in a final model for the e-Puck architecture in Webots Simulator application. 10^3 simulations are used (in boxplot measurements) to evaluate each model, and the completed tasks number CT considered in this work is an average according to these simulations. Fig. 11 shows an improvement and an increase in the total tasks number completed in E_1 , according to the average and median of the result of tasks completed by the robot team as the amount of robots increases. This factor can be seen in Fig. 11(a), for GIACA, and in Fig. 11(b), for GSTIACA. In addition, one of the additional model features is the Tabu queue usage for GSTIACA. This improvement is due to the fact that the robot is able to check its queue and also other robots queue, avoiding recently traveled paths. To take as an example, the optimized model GSTIACA using $N = 3$ robots completes $CT_3 = 43.41$ visits in all rooms, whereas for the optimized model GIACA, only $CT_3 = 17.89$ are completed. Which means that the GIACA model is effective at 0.41% compared to the model proposed in this GSTIACA work, with $N = 3$ and simulated at E_1 .

Similar results can be observed for the environment E_2 of Fig. 12 and Table 2, that is, this means that as the robots number $N = \{1, 2, 3, 4, 5, 6\}$ increases, more CT are reached by the model. For example, for $N = 3$, Fig. 12(a) shows GIACA with a total of $CT_3 = 17.26$ and for Fig. 12(b) we have a total of $CT_3 = 39.33$. This also means that adding the queues and sharing them between the robots causes a significant improvement in the effectiveness of the GSTIACA model in relation to the GIACA. Additionally, we can say that on average, the E_2 environment is more complex than E_1 due to E_2 having fewer tasks completed CT for simulations for any robots number and regardless of whether the model is GIACA or GSTIACA.

For the E_3 environment, a similar experiment was performed and is shown in Fig. 13 and Table 2. In this case, Fig. 13(a) presents the GIACA results, while Fig. 13(b) presents GSTIACA results. It is possible to see that in relation to the previous models E_3 can be considered the most complex environment, since for all teams sizes N we have that the completed tasks amount CT is always lower, both for GIACA and GSTIACA.

Finally, for the E_4 environment, the results are shown in Fig. 14 and Table 2. For GIACA shown in Fig. 14(a) with $N = 3$ robots we have a total of $CT_3 = 14.78$, for GSTIACA in Fig. 14(b) we have a total of $CT_3 = 34.21$.

Table 2
Table with boxplots values shown in Figs. 11, 12, 13 and 14.

Model	Map	Characteristic	N1	N2	N3	N4	N5	N6	Average
GIACA	E_1	Upper whisker	7.00	15.00	27.00	44.00	66.00	97.00	42.667
		3rd quartile	4.00	11.00	20.00	33.00	50.00	75.00	32.167
		Median	3.00	10.00	18.00	29.00	44.00	66.00	28.333
		1st quartile	2.00	8.00	15.00	25.00	39.00	60.00	24.833
		Lower whisker	0.00	4.00	8.00	14.00	23.00	38.00	14.500
		Mean	3.39	9.48	17.89	28.86	44.47	67.02	28.518
GSTIACA	E_1	Upper whisker	12.00	32.00	57.00	89.00	126.00	186.00	83.667
		3rd quartile	9.00	25.00	47.00	74.00	108.00	158.50	70.250
		Median	8.00	23.00	43.00	69.00	102.00	149.00	65.667
		1st quartile	7.00	20.00	40.00	64.00	96.00	140.00	61.167
		Lower whisker	4.00	13.00	30.00	50.00	79.00	113.00	48.167
		Mean	7.92	22.95	43.41	68.92	102.13	149.43	65.793
GIACA	E_2	Upper whisker	7.00	17.00	27.00	41.00	64.00	88.00	40.667
		3rd quartile	4.00	11.00	20.00	31.00	48.00	69.00	30.500
		Median	3.00	9.00	17.00	28.00	42.00	62.00	26.833
		1st quartile	2.00	7.00	15.00	24.00	37.00	56.00	23.500
		Lower whisker	0.00	2.00	9.00	14.00	22.00	37.00	14.000
		Mean	3.42	9.19	17.26	27.89	42.44	62.51	27.118
GSTIACA	E_2	Upper whisker	12.00	29.00	53.00	80.00	114.00	166.00	75.667
		3rd quartile	9.00	23.00	43.00	67.00	98.00	143.00	63.833
		Median	8.00	21.00	39.50	63.00	92.00	135.00	59.750
		1st quartile	7.00	19.00	36.00	58.00	87.00	127.00	55.667
		Lower whisker	4.00	13.00	26.00	45.00	71.00	103.00	43.667
		Mean	7.86	21.09	39.33	62.51	92.49	134.84	59.687
GIACA	E_3	Upper whisker	7.00	13.00	24.00	37.00	52.00	79.00	35.333
		3rd quartile	4.00	9.00	17.00	27.00	40.00	58.00	25.833
		Median	3.00	8.00	14.00	23.00	35.00	51.00	22.333
		1st quartile	2.00	6.00	12.00	20.00	31.00	44.00	19.167
		Lower whisker	0.00	2.00	6.00	10.00	18.00	26.00	10.333
		Mean	3.27	7.93	14.38	23.50	35.30	51.51	22.648
GSTIACA	E_3	Upper whisker	13.00	26.00	47.00	72.00	105.00	143.00	67.667
		3rd quartile	9.00	20.00	37.00	59.00	87.00	122.00	55.667
		Median	7.00	18.00	34.00	55.00	81.00	115.00	51.667
		1st quartile	6.00	16.00	30.00	50.00	75.00	108.00	47.500
		Lower whisker	2.00	10.00	21.00	38.00	57.00	88.00	36.000
		Mean	7.40	18.19	33.87	54.93	80.88	115.22	51.748
GIACA	E_4	Upper whisker	7.00	16.00	24.00	37.00	53.00	77.00	35.667
		3rd quartile	4.00	10.00	17.00	27.00	40.00	58.00	26.000
		Median	3.00	8.00	15.00	23.00	35.00	51.00	22.500
		1st quartile	2.00	6.00	12.00	20.00	31.00	45.00	19.333
		Lower whisker	0.00	2.00	5.00	10.00	18.00	26.00	10.167
		Mean	3.22	8.03	14.78	23.42	35.27	51.16	22.647
GSTIACA	E_4	Upper whisker	13.00	28.00	46.00	71.00	104.00	143.00	67.500
		3rd quartile	9.00	21.00	37.00	59.00	86.00	122.00	55.667
		Median	7.00	19.00	34.00	55.00	80.00	114.00	51.500
		1st quartile	6.00	16.00	31.00	51.00	74.00	108.00	47.667
		Lower whisker	2.00	10.00	22.00	39.00	57.00	87.00	36.167
		Mean	7.42	18.69	34.21	55.15	79.92	114.54	51.655

Thus, we can conclude that E_4 is also the most complex environment for the surveillance task, even though all of them have the same rooms number, E_4 proved to be the most complex among $\{E_1, E_2\}$. However, when we compare the average for all robot numbers N , we realize that in average, the environment, being complex allows a task points number less than $\{E_1, E_2\}$.

Table 2 summarizes the results for all environments from E_1 to E_4 in relation to GIACA and GSTIACA models and with a range of $N = 1, 2, 3, \dots, 6$ robots. From the analysis, the environment with the lowest difficulty level is E_1 (for GIACA and GSTIACA), and with the highest level is the environment E_3 (for GIACA and GSTIACA), which was considered on average as the most complex. So we have $E_1 < E_2 < E_4 < E_3$ ordered depending on the difficulty level for completed tasks. All environments have the same number of 7 doors and 7 rooms, but even so, there is actually a difference in difficulty compared to them. Some environments are more hostile and difficult than other due to several features, such as, distances between doors, some rooms are more isolated (rooms with only one exit) in relation to others. In this case, another issue raised was the number of rooms with at least 3 outputs, in this case, the E_1 environment has 2 rooms with 3 doors (the easiest for patrolling), as it avoids jam and bottleneck phenomena among robots, the same phenomenon that occurs in pedestrian evacuation works [30,31,36]. Environments E_2 and E_4 have the same distribution, but there are big rooms with only 1 door, these being of intermediate difficulty. For these intermediate environments, the more

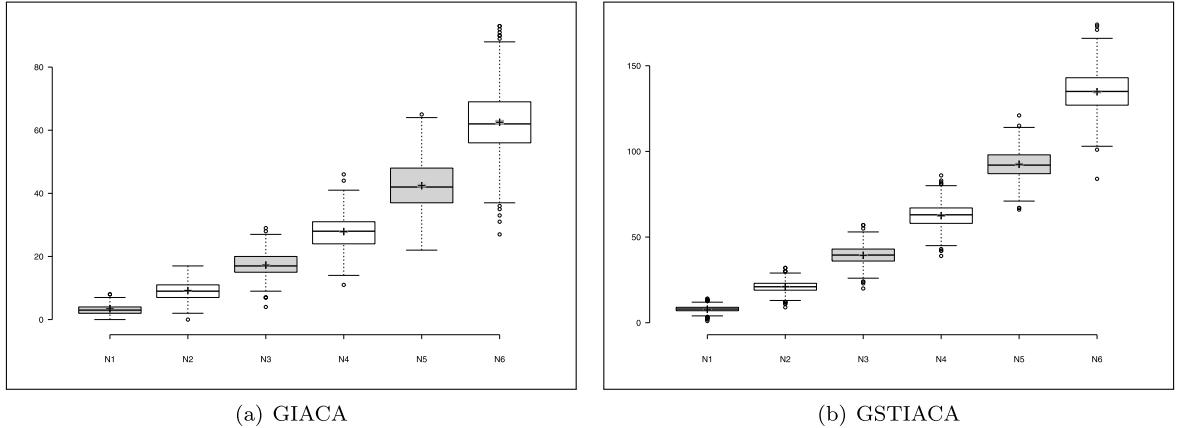


Fig. 12. Boxplots showing average visits for each $N = \{1, 2, 3, 4, 5, 6\}$ robots. E_2 environment.

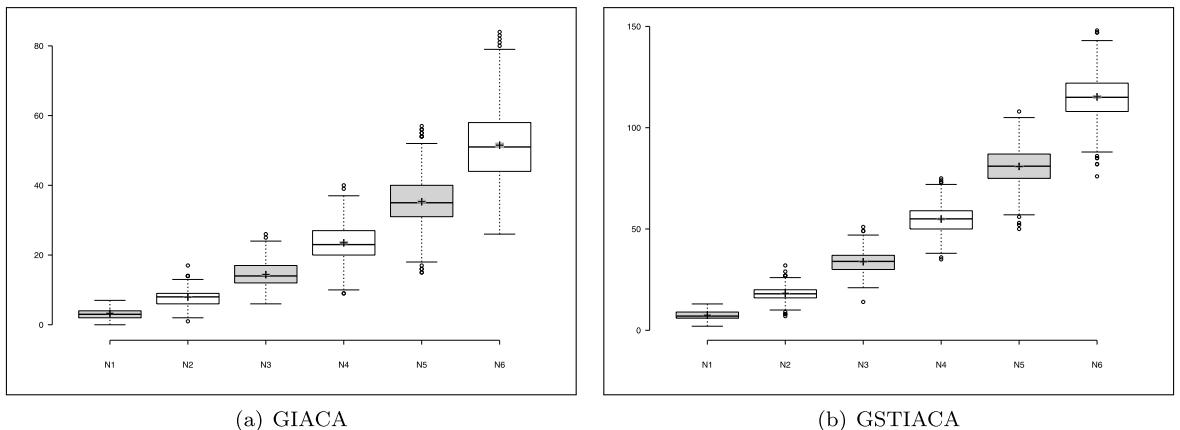


Fig. 13. Boxplots showing average visits for each $N = \{1, 2, 3, 4, 5, 6\}$ robots. E_3 environment.

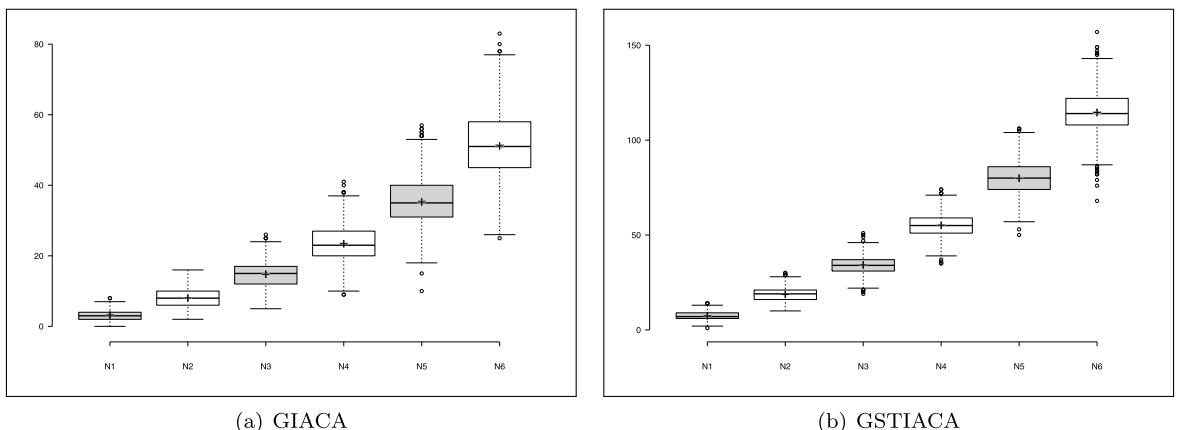


Fig. 14. Boxplots showing average visits for each $N = \{1, 2, 3, 4, 5, 6\}$ robots. E_4 environment.

proportionally the environment is divided in terms of area per room, the easier it becomes for robots to patrol, so in this case we can say that E_2 is simpler than E_4 . Finally, The environment E_3 has 1 single room very big in comparison to others, which is the most difficult among the 4 evaluated environments. Another important factor is balanced by the robots number, because when its

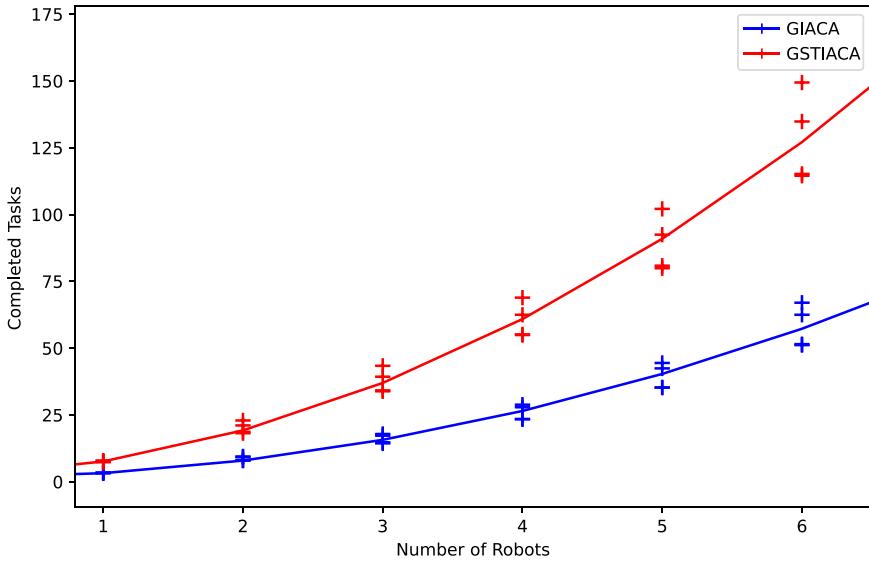


Fig. 15. Comparison between GIACA model, represented by (■), and GSTIACA model, represented by (+), using polynomial regression in the environments E_1 , E_2 , E_3 and E_4 . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

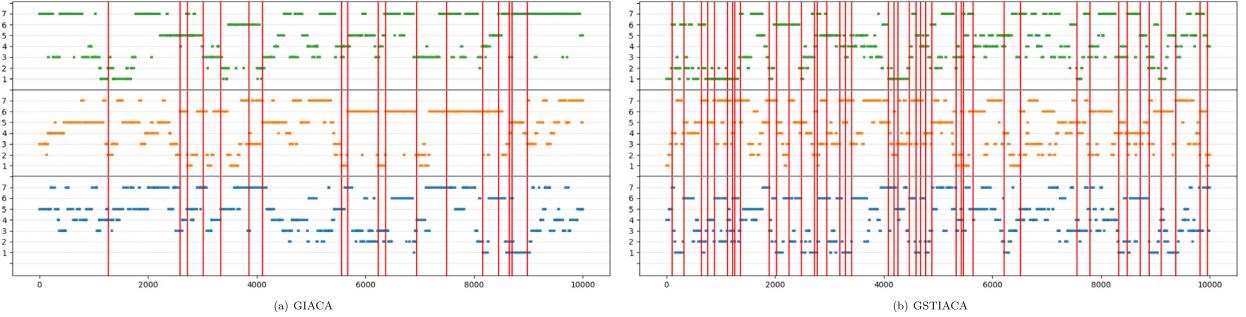


Fig. 16. Representation of the tasks completed number by the two models GIACA and GSTIACA over simulations lasting 1000 steps, with (●) representing each task point, (■) represents the robot p_1 , (○) represents the robot p_2 and (▲) represents the p_3 robot. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

increases, the completed tasks number C_T also increases with this factor. This feature also increases the hardware cost, because the increase in the robots number employed in the system also increases the synchronization cost and robotic agents control.

4.3.2. Data analysis of the simulations regressions

The regression, shown in Fig. 15, describes the best results achieved through optimization, comparing the GIACA and GSTIACA models of the Table 2, using the quantity $N = \{1, 2, 3, 4, 5, 6\}$ robots, showing the tasks performed number. The graphs show a regression analysis using the Ordinary Least Squares method, which predicts the amount of y_{CT} tasks performed (y-axis) according to the number of x_N robots used (x-axis) in both GIACA and GSTIACA. The graphic symbols represented by (+), for GIACA, and by (+), for GSTIACA, present the experimental values achieved in the four environments: $\{E_1, E_2, E_3, E_4\}$, in this case, with 10^3 simulations for each environment. The lines represented by the symbols (-), for GIACA, and by (-), for GSTIACA, present the polynomial regressions values obtained through the experimental results. From the experimental values, a polynomial regression was obtained, which resulted in the polynomial, represented by the symbol (-), for the GIACA model according to Eq. (6). The value y_{CT} is known to represent the CT amount in relation to the number x_N robot growth.

$$y_{CT} = 1.525x_N^2 + 0.1361x_N + 1.565 \quad (6)$$

As for GSTIACA, we have Eq. (7) that represents the growth curve (-) of the tasks number completed y_{CT} as the robots number x_N increases.

$$y_{CT} = 3.070x_N^2 + 2.412x_N + 2.129 \quad (7)$$

Thus, it is possible to see that for all environments, GSTIACA performs better than GIACA. In addition, the curve indicates that the GSTIACA model shows faster asymptotic growth, compared to the GIACA, as the robots number increases. That corroborates that the new model proposed here, GSTIACA, is better than its precursor GIACA, which in turn is better than TIACA [75], which is better than IACA [69], since here we got the improved values. Additionally, for GSTIACA, even using only $N = 1$ robot, it is possible to notice that in all cases the lower whisker of the interval managed at least $CT = 2$ in $T = 10^4$ steps. On the other hand, regarding this value, in GIACA there were some lower whiskers in which $CT = 0$, cases in which no task was completed. That is, even if there is no queues sharing (a single robot), it is possible to observe that only the Tabu search adoption linked to the CA model was able to significantly improve the GSTIACA model.

4.3.3. Performance analysis using task points

Task point is a metric used in many robotics works, such as [6, 49, 69, 74], where the robot team performance is calculated as a function of the number of complete visits to all rooms in the environment. That is, when at least one robot passes through each of the rooms, we can say that a task point is performed. The sum of all task points gives us the CT metric, which are completed tasks, which measures the total task points in 10000 iterations that were performed by the robots. The task points analysis is represented in a graphic shown in Fig. 16, on the x-axis we have represented the time $T = 10^4$ and the line (●) represents each task point, that is, each task completed. The y-axis represents each $N = 3$ robots being represented by a different color, with (■) representing the p_1 robot, (□) represents the robot p_2 and (■) represents the robot p_3 . These colors also signal the room in which the robot is at that moment. In addition, on the y-axis each room is represented by a number between $\Omega_i = \{1, 2, 3, 4, 5, 6, 7\}$. Each robot has visits to the rooms represented separately at a time $t \in T$, and a Task Point (●) occurs each time all rooms are visited at least once time by the robots team. After that, the visit count is restarted and the process is repeated. This performance analysis allows to have an idea of the environment exploration efficiency over the simulation time. Thus, in a simulation selected for the task points analysis, Fig. 16(a), represents the GIACA model with $N = 3$ robots, and we obtained $CT = 18$. On the other hand, in Fig. 16(b) represents GSTIACA, using the same robots number as in GIACA, but in this case we obtained $CT_3 = 42$ complete surveillance cycles.

Another analysis that can be done is in relation to the t moment when the first task point was completed for both models. In GIACA we have that at $t_s = 1273$ it was the initial moment that the first task was completed and $t_f = 8973$ was the final moment that the last task was completed. For GSTIACA we have that at $t_s = 101$ it was the initial moment that the first task was completed and $t_f = 9950$ was the final moment that the last task was completed. Thus, the workflow rate can be calculated according to the following Eq. (8). First, $\frac{CT}{N}$ represents the amount that each robot in the system contributed to the surveillance task. The fraction $\frac{|t_f - t_s|}{T}$ represents the working time percentage.

$$\text{flow} = \frac{|t_f - t_s| \times CT}{T \times N} \quad (8)$$

Thus, according to Eq. (8), where $T = 10^4$, $N = 3$ and $CT = 18$ the work flow performed at GIACA was $\text{flow} = 4.62$ patrols per time interval. In GSTIACA we see that the flow was $\text{flow} = 13.79$ per time interval with $T = 10^4$, $N = 3$ and $CT = 42$. Which means that the team's performance in GSTIACA was ≈ 2.98 times better than that observed in GIACA. Thus, it is possible to observe the gain in terms of using the workforce in the GSTIACA model, since in GIACA there is an under-use of robotic agents. This is because it is observed some robots rework instead of looking for rooms not recently visited.

4.3.4. Visual analysis of environmental coverage

These simulations used for this experiment are called storyboards, representing an illustrative version of how the simulations occurred over time and demonstrate the pheromone and step map for the GIACA and GSTIACA simulation models. Fig. 17 contains simulations examples running over time with $t \in T$ and the storyboards printed during $t = \{2000, 4000, 6000, 8000, 10000\}$ in the environment E_1 , and represents the pheromone maps and the environment coverage, measured by the step map. The pheromone map is represented within a temperature color scale that ranges from the coldest Ph_b , represented by the color (■) for low pheromone deposit rates, to the hottest Ph_r , represented by the color (■) for high pheromone rates. The stepmap follows the same color scale as the pheromone map, but the blue SM_b represents few steps taken in that cell, while the reddish colors SM_r represents many steps taken per cell.

Figs. 17(a), 17(e), 17(i), 17(m) and 17(q) represent the pheromone maps for GIACA. In GSTIACA we have the pheromone map simulations represented by Figs. 17(c), 17(g), 17(k), 17(o) e 17(s). In this case, the pheromones distribution gives an idea of how pheromones influence the robots movement. In the GIACA model it is possible to observe that the pheromone has a larger radius and greater durability, as it leaves a larger trail. In the GSTIACA model, it is possible to observe that the pheromone is more concentrated around the robot, having less durability, but this factor is compensated by the queues usage, which helps in a greater robots spreading throughout the environment. Indicating that only a GIACA model based only on indirect communication from robots is not enough to guarantee that the surveillance task is carried out efficiently. Just like the ants [41], there is also a need for direct communication between the robotic agents to guarantee a good robot team performance for the patrolling task.

The variables used for indirect communication that represent the deposit (δ, σ) and the evaporation or decline (β) interact in a way that the maps were arranged in this way. For example, in GIACA, we have $\delta = 0.163$ and $\sigma = 0.065$ which are values for the pheromone deposit are directly linked to the evaporation rate $\beta = 0.061$, taking into account that the maximum value used is $\tau_{max} = 38.162$. For GSTIACA we have $\delta = 0.859$, $\sigma = 0.016$ and $\beta = 0.322$, which are even higher. These results in a smaller pheromone trail in the environment (less shared global memory), since the maximum pheromone rate does not exceed $\tau_{max} = 30.364$, lower than in GIACA. Thus, the pheromone map images can be explained considering the analysis of these parameters.

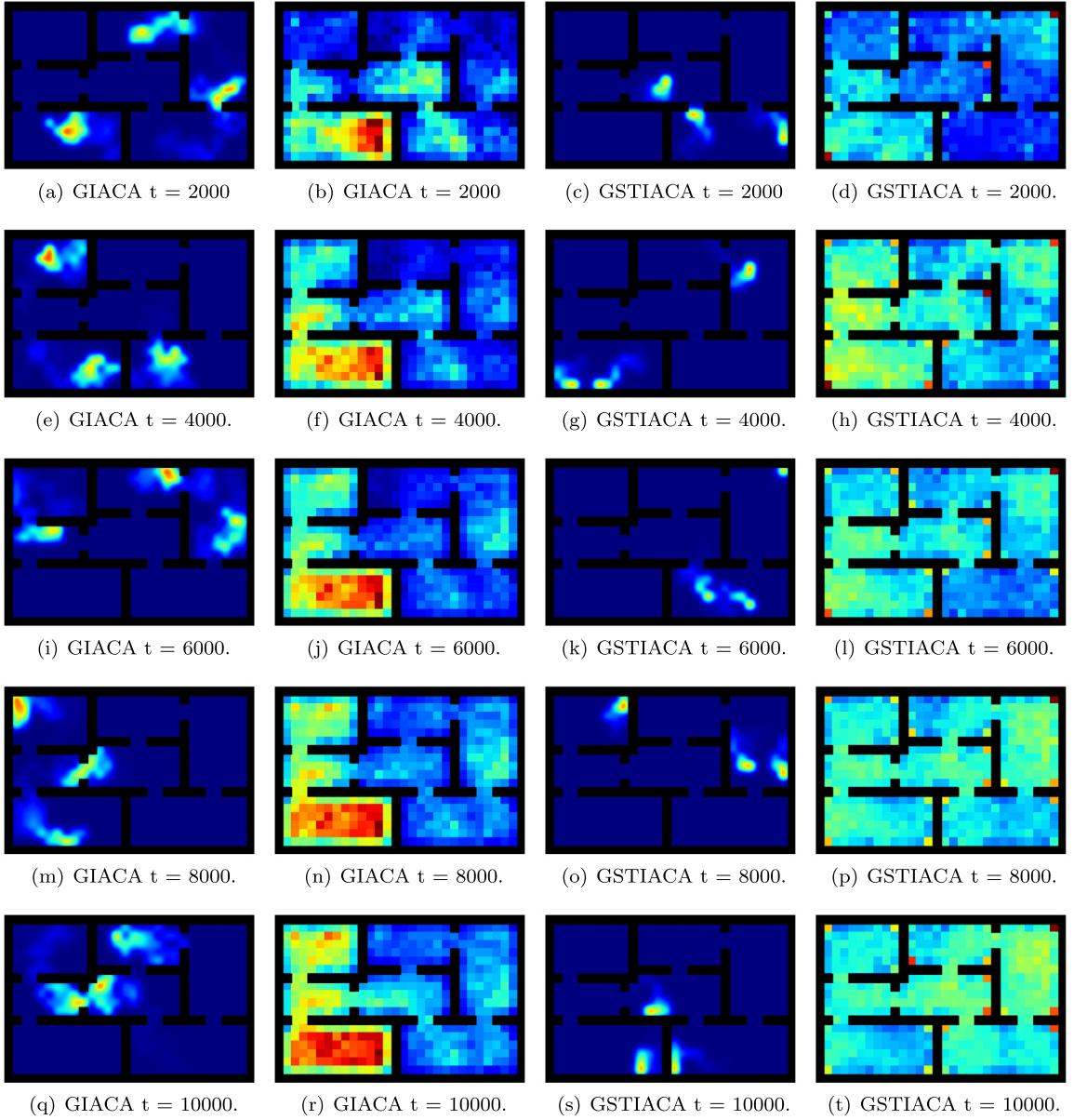


Fig. 17. Storyboards (simulation over time) that represents pheromone maps GIACA and GSTIACA; and GIACA and GSTIACA step maps (c, g, k, o, s), using $N = 3$ robots, where the scale varies from (■), representing the least visited areas, to (■), representing the most visited areas. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

For the results presentation for each storyboard we will use a tuple in which the ordered pair is (Ph_b, Ph_r) , in which we present the maximum and minimum values reached by the pheromone levels. For GIACA Fig. 17(a) has (0.0, 12.22). Fig. 17(e) has (0.0, 11.19). Fig. 17(i) has (0.0, 11.77). Fig. 17(m) has (0.0, 12.80) and Fig. 17(q) has (0.0, 13.29) values. For GSTIACA Fig. 17(c) has (0.0, 26.09). Fig. 17(g) has (0.0, 26.09). Fig. 17(k) has (0.0, 26.09). Fig. 17(o) has (0.0, 28.84) and Fig. 17(s) has (0.0, 26.09) values.

The stepmap in Fig. 17 represents a map in which during the $T = 10^4$ steps a matrix identical to the simulation environment is updated and each time a robot passes over a cell, the (+1) value is added to the matrix. Initially, all values in the step matrix are set to (0). Figs. 17(b), 17(f), 17(j), 17(n) and 17(r) represent the step maps for GIACA. In GSTIACA we have the step map simulations represented by Figs. 17(d), 17(h), 17(l), 17(p) and 17(t). For the results presentation referring to each storyboard we will use a tuple in which the ordered pair (SM_b, SM_r) . For GIACA Fig. 17(b) has (0, 175). Fig. 17(f) has (0, 53). Fig. 17(j) has (0, 94). Fig. 17(n) has (0, 133) and Fig. 17(r) has (0, 151) values. For GSTIACA Fig. 17(d) has (0, 167). Fig. 17(h) has (0, 53). Fig. 17(l) has (0, 68). Fig. 17(p) has (0, 115) and Fig. 17(t) has (0, 139) values.

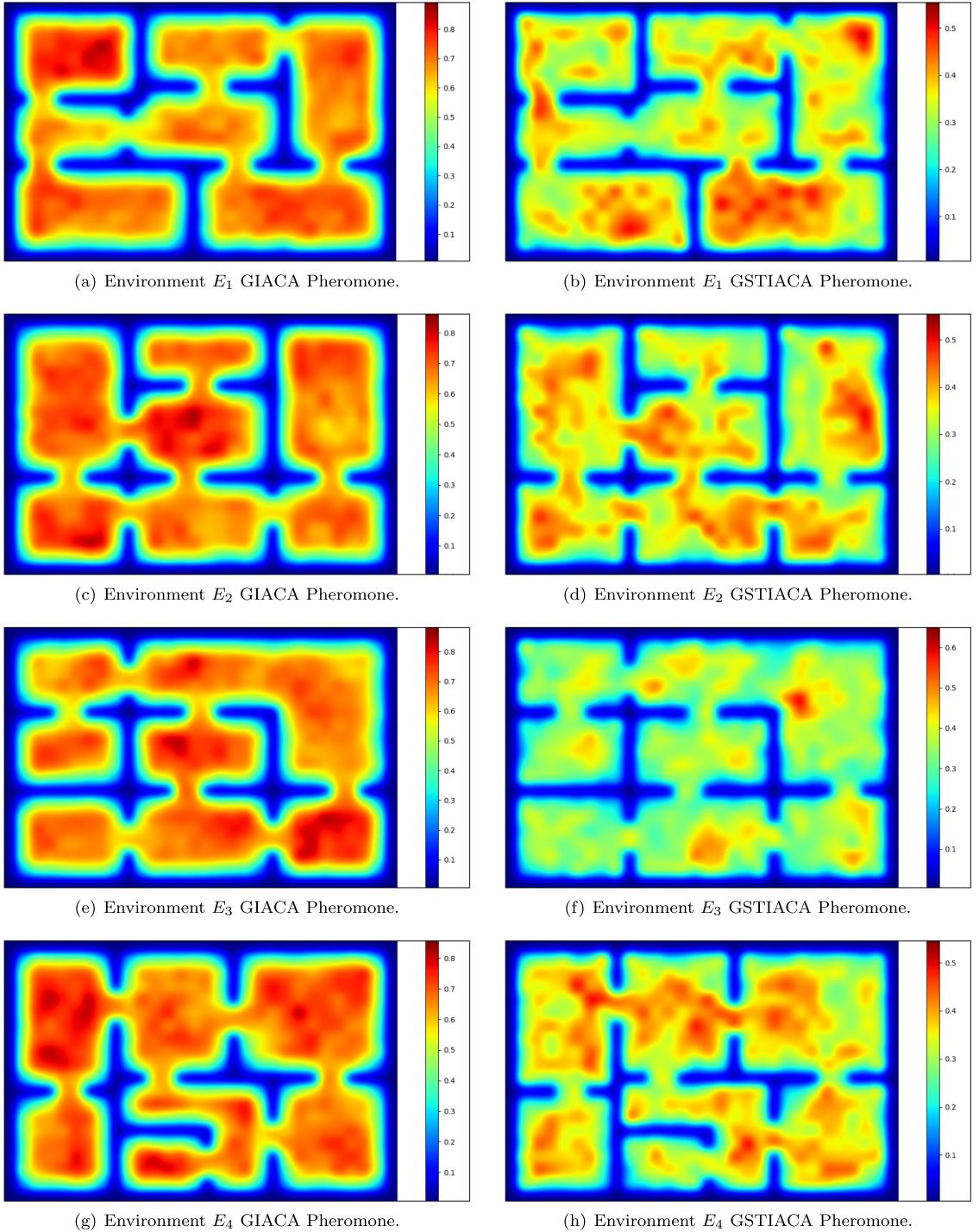


Fig. 18. Experiments to verify the mean pheromones rate over the environments $\{E_1, E_2, E_3, E_4\}$, using $N = 3$ robots, where the scale varies from (■), representing the least visited areas, to (■), representing the most visited areas. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

From these results it is possible to conclude that robots explore the environment more uniformly when it comes to GSTIACA, compared to the GIACA algorithm. In GIACA it is possible to observe an agglomeration in the 3 rooms on the left, and in GSTIACA a more uniform robots distribution throughout the rooms, in a more homogeneous way. GSTIACA also leaves a smaller pheromone

trail, which may indicate that it is making better usage of the information provided by the Tabu line and giving priority to this over the pheromones usage.

4.3.5. Experiments covering the environment: pheromone distribution and step maps

This section describes experiments performed to quantify the robots interaction with the environments E_1 , E_2 , E_3 and E_4 . These experiments, described in Figs. 18 and 19 represent the pheromone maps and the step maps, respectively. The pheromone map describes the accumulated pheromone distribution, in a process similar to the step map, in this case, considering the final pheromones result in the map after $T = 10^4$, then each accumulated value in the matrix is divided by 10^3 , which represents the times number the experiment is repeated, where each cell value is calculated according to an average of 1000 simulations. In the environments map images, the colder colors, represented by (■) indicate that during the 10^3 simulations, those spaces were less visited, while the warmer colors were represented by (■), indicate those areas that were visited a lot. The pheromone and step experiments, consider the intermediate colors, such as, (■ ■ ■ ■ ■), represent the intermediate values between the least visited areas and the most visited areas. Figs. 18(a), 18(c), 18(e) and 18(g) represent the GIACA model pheromones distribution. Figs. 18(b), 18(d), 18(f) and 18(h) show the GSTIACA model pheromones distribution. Some differences between the GIACA and GSTIACA models are noticeable when compared to the pheromone distributions on the maps. From there it is possible to have an idea of how the pheromones spread according to the map geometry. It is possible to perceive from the images to measure the pheromone spread throughout the environment that GSTIACA has a greater heterogeneity in the pheromone spread in contrast to GIACA, which has a more homogeneous spread, concentrated in specific locations in the environment. Therefore, this is another characteristic that puts GSTIACA with better performance for the robot team than GIACA. If the environment becomes very homogeneous, the difference between the pheromones is low, which makes the system not provide additional information to the agents and makes this system on the verge of randomness. Besides that, it means that GSTIACA, for having in its intrinsic nature the use of shared memory, ends up using less indirect communication between robots and using a greater degree of direct communication [41].

Figs. 19(a), 19(c), 19(e) and 19(g) report the GIACA model steps layout. Finally, Figs. 19(b), 19(d), 19(f) and 19(h) demonstrate the GSTIACA model steps distribution. The figures step maps represent the steps number, that is, the steps sum that the robots remained in each lattice CA cell. In order to improve the results measurement accuracy, 10^3 simulations were made, and the steps number in each cell is the steps average in each cell during 10^3 simulations. With these results it is possible to measure the robots' movement by the map extent. The GIACA model has a greater robots concentration in the central part of the CA grid. On the other hand, GSTIACA can make it stay in deadlock near the corners because of the Tabu search. However, GA optimized for the last queue element to be dequeued, generating this accumulation close to the corners. However, even with this characteristic, it is still the model that presented the largest complete tasks in relation to GIACA, TIACA [94] and IACA [69], which indicates that the model is suitable to be implemented in a real environment using a robotic architecture.

With this experiment it is possible to conclude that even if GSTIACA presents a possible robots congestion near perpendicular obstacle areas (⊥), it is still possible to say that it is superior in relation to GIACA, due to the steps being smaller, similar phenomena tend to occur in work on pedestrian evacuation [30,31,36]. Thus, this can be a characteristic to be observed in later studies. For the experiment with pheromones, as the $\tau_{max} = 38.162$ value of GIACA allow to accumulate more pheromone by simulation, this means that the values (■, ■) is more prevalent. For the GSTIACA model, the values (■, ■, ■) predominate on pheromone maps, since $\tau_{max} = 30.364$ has lower values.

4.4. Experiments using e-Puck robot in the webots simulation environment

In this section, some experiments will be presented on the GIACA and GSTIACA models implementation in the Webots simulator using the e-Puck robotic architecture. These experiments aim to corroborate that the two surrogates GIACA and GSTIACA proposed here will serve as a model for implementation in the real world, using not only e-Puck, but any other mobile robotic architecture.

4.4.1. Setting up the simulation environment on Webots

A brief materials description used for the experiments in the Webots simulator, described in [95], using the e-Puck [84] architecture, will be presented. The Webots version used was R2019b Revision 1. To emulate the pheromone evaporation in Webots, a Pen-type node was used, inside a Slot node, which is a tool that allows you to draw a trail over a texture. The texture represents the CA lattice, with cells identical to each other and serves to guide the programmer regarding the robot's movement. The positioning of this tool provides a robots movement notion, thus allowing to have an idea of the pheromones distribution and steps distribution on the map. The evaporation rate used to make the pen slot illustration on Webots was $\beta = 0.000606$ for GIACA and $\beta = 0.00322$ for GSTIACA, which are rates chosen in proportion $\frac{1}{100}$ in relation to the discretized values found by GA. These values are in line with that presented in previous experiments, which was $\beta = 0.061$ for GIACA (which was rounded to 3 decimal places) and $\beta = 0.322$ for GSTIACA. The size of each cell in the lattice is 14×14 centimeters, about 7 centimeters larger than the e-Puck robot diameter, which is 3.5 centimeters in radius. The E_1 environment has been increased and has 40×60 cells, this increase was due to the virtual obstacles creation [96], since they block the doors of very small environments.

In order to describe the synchronization and parallel localization approach proposed in [67], it is known that robots need to perform their movements at the same time, as it is a CA-based control. In order to simultaneously synchronize the robot team movement, global information was shared between the robots through a [67] server. The synchronization messages were sent by the robots using text files (.txt), which contain the position in the lattice and the robot time t . Each agent sends its t time to the server. The server is responsible for reading and processing the next cluster movement in the time $t + 1$. All existing conflicts

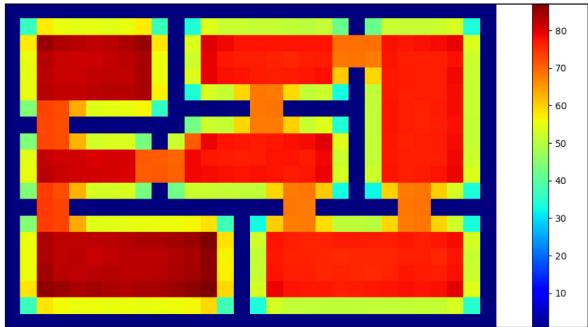
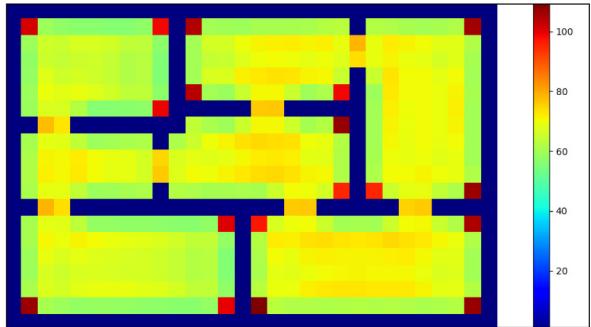
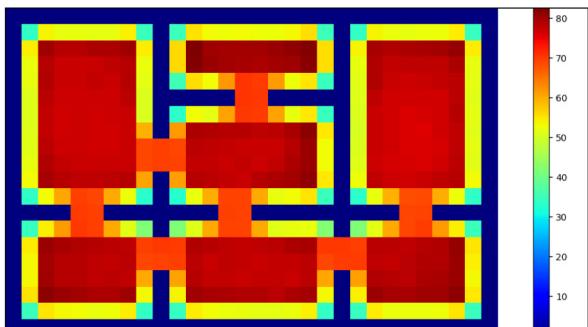
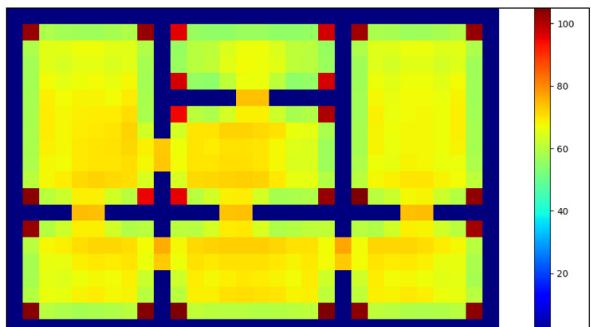
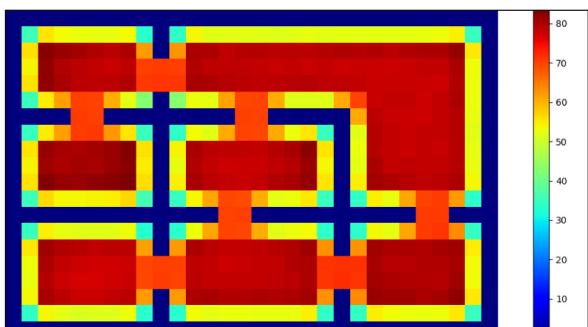
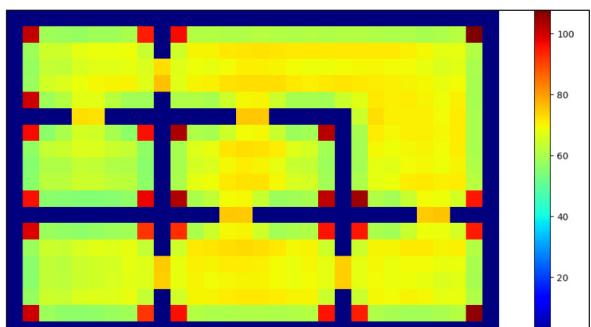
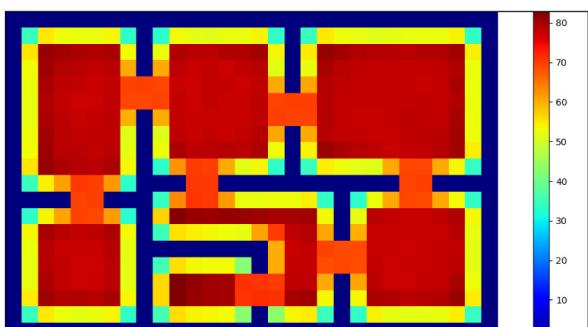
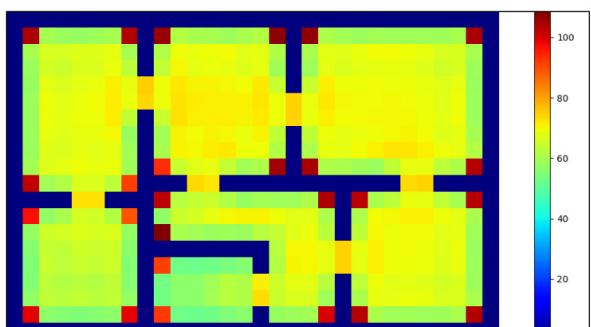
(a) Environment E_1 GIACA Steps.(b) Environment E_1 GSTIACA Steps.(c) Environment E_2 GIACA Steps.(d) Environment E_2 GSTIACA Steps.(e) Environment E_3 GIACA Steps.(f) Environment E_3 GSTIACA Steps.(g) Environment E_4 GIACA Steps.(h) Environment E_4 GSTIACA Steps.

Fig. 19. Experiments to check the average step maps rate across environments $\{E_1, E_2, E_3, E_4\}$ using $N = 3$ robots, where the scale ranges from (■), representing less visited areas, up to (■), representing most visited areas. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

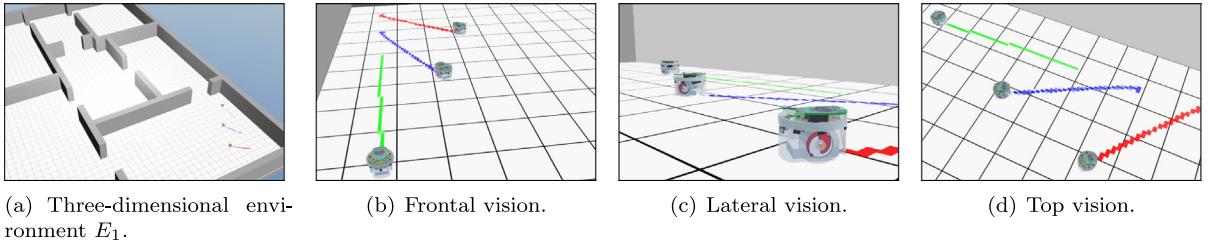


Fig. 20. Pheromones maps for GIACA and GSTIACA models using an environment modification E_1 with 40×60 cells, where p_1 is represented by (■), p_2 is represented by (■) and p_3 is represented by (■). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

are solved by the server. The server in turn sends the message back to the robots, indicating its next movement position [44]. Each swarm's robots member read in parallel and independently of the server's file, search for the robot's number and execute its movement. The server continuously reads the files of each robot during its movement. The server is responsible for checking at every moment whether each robot has finished its movement. The robot indicates its movement when recording the time $t+1$ in its file. If time information $t+1$ is recorded in all files, the server will calculate the new robot position. The GPS usage has improved the location accuracy, in relation to the odometry technique [19,97], using a GPS-type Node which is used to improve refinement and accuracy location [67]. In addition, a turning angle of the saved robot and an unsaved one were considered to improve the robot performance from the [67] team. In addition to the GPS Node, a Compass-type node is also used, which represents a compass and allows to obtain from the environment the angle inclination information in which the robot is at the present moment. Using information from the compass, GPS, the location discretized by the cellular automaton at the time t , the location at the cellular automaton at the time $t+1$ and the estimated destination position, it is possible to calculate the robot's movement in real time, using as much of all the data that can be obtained to correct its movement as it obeys the server commands.

When it comes to the Webots simulation environment, the E_1 environment was expanded to 40×60 cells, each cell is 14 cm, twice the diameter of the e-Puck which is 7 cm. Therefore, the actual size of the room is 840 cm \times 560 cm. In this case, a plane is inserted that simulates the CA cells identical to each other, with the same dimension and square shape. Subsequently, obstacles are created and a height for them is configured. The obstacles present on the map are treated as virtual obstacles and expanded [96] to avoid perpendicular collisions (⊥) in corners with the walls that delimit the environment. Then, $N = 3$ robots are added to the environment with a stepping pen, where p_1 is represented by (■), p_2 is represented by (■) and p_3 is represented by (■). This pen draws on the texture present in the plane that has the CA cells. Fig. 20(a) shows the environment with the three robots initially in the same room. These robots disperse in accordance with their respective controllers that serve as a surrogate for application in the real world.

Fig. 20 presents three simulation environment views with $N = 3$ robots. Fig. 20(b) presents the three robots in movement in a frontal view. The second image in Fig. 20(c) shows the three robots preview and Fig. 20(d) shows the simulation top side preview. All of these visions helped us to generate the best possible environment configuration using the robotic e-Puck architecture within the Webots simulation environment. Additionally, it was from these configurations that it was possible to generate the experiments that will be explained in the next section.

4.4.2. Surveillance task simulation using e-puck

This section describes the GIACA and GSTIACA models results using the e-Puck [98] architecture in the Webots [95] simulation environment. These experiments using a real simulation environment reinforce that the models proposed here can be applied in a real context using other robotic architectures types, but especially e-Puck, which was tried and tested by us here. Initially, the results will be presented visually with $N = 3$ robots. First, we take pictures every $t = 10$ step and store those pictures on a computer. Later, we chose some more representative photos in order to give a sample of how the experiment was carried out and evaluated. First, a test with 3 robots was used to test the pheromone deposit during the simulation. Fig. 21 shows the screenshot in the instant $t = 1000$ in which the robots use an evaporative pen to paint the floor, indicating that they have been there recently. Fig. 21(a) shows the pheromone manages to stay in the GIACA environment for longer, because the values $(\delta, \sigma, \beta, \tau_{max})$ present this characteristic, as presented in the previous sections. Unlike the GSTIACA model in which the genetic algorithm, observing the other model parameters $(\overline{CQ}, QP_\otimes, QP_\emptyset, |Q|)$ evolved these characteristics to values $(\delta, \sigma, \beta, \tau_{max})$, which is also visible in the image in Fig. 21(b). It is possible to notice that GIACA tends to avoid less efficiently that two robots remain in the same room, in this case, it is possible to notice that p_1 and p_3 are in the same room. Unlike GSTIACA in which each robot p_1 , p_2 and p_3 are in different rooms giving greater efficiency in patrolling the environment.

Fig. 22 represents the graphic covering the environment from the step map measurement. First, Fig. 22(a) shows the GIACA coverage evolution, in which at time $t = 1000$ the model with $N = 3$ robots managed to cover only 3 rooms out of the 7 possible rooms (42% of the environment). On the other hand, Fig. 21(b) for GSTIACA it is possible to notice that the set of $N = 3$ robots achieved coverage of 4 of the 7 rooms of E_1 (57% of the environment). Which again means that the GSTIACA parameters really converge to more efficient simulation environment coverage.

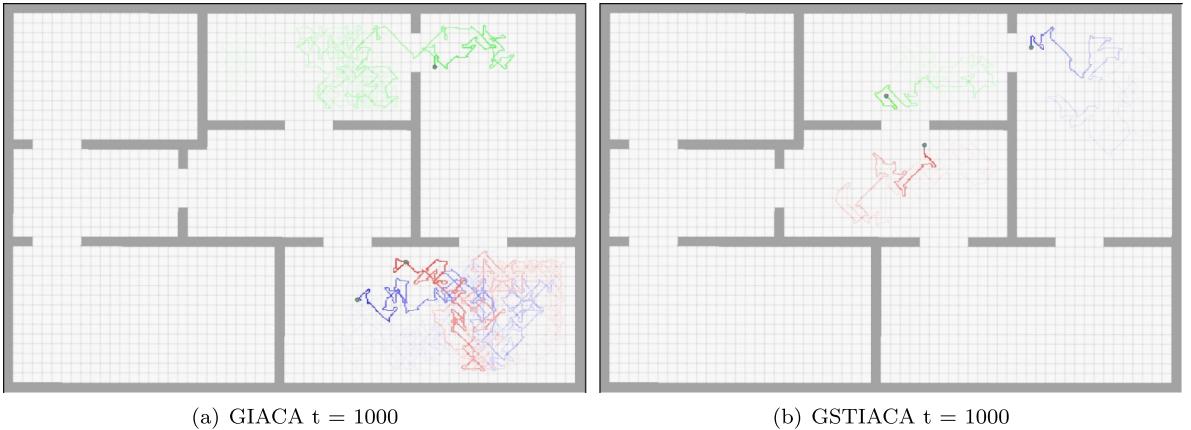


Fig. 21. Pheromone maps for GIACA and GSTIACA using a modification of environment E_1 with 60×80 cells, where p_1 is represented by (■), p_2 is represented by (□) and p_3 is represented by (■), in $t = 1000$ steps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

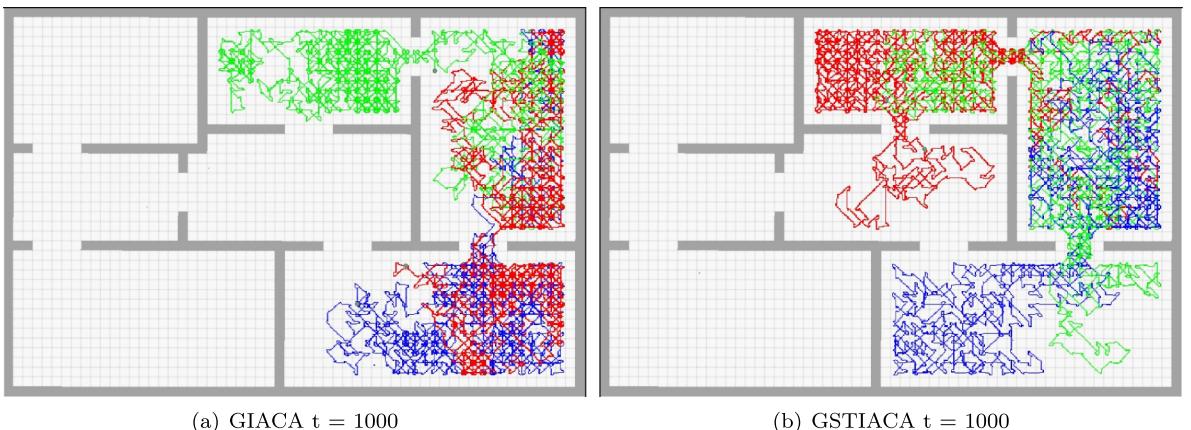


Fig. 22. Steps maps for GIACA and GSTIACA using a modification of environment E_1 with 60×80 cells, where p_1 is represented by (■), p_2 is represented by (□) and p_3 is represented by (■), in $t = 1000$ steps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 23 presents some experiments carried out in the Webots robotic simulator with the e-Puck architecture, to test and measure the final model with the optimized parameters, measuring step quantities and pheromone distribution. We have for this set of experiments an evaluation for $N = 3$ robots during $T = 10^4$ time steps, $t \in T$, where $t = \{2000, 4000, 6000, 8000, 10000\}$. These experiments are the simulation continuation with $t = 1000$ of Figs. 21 and 22. The experiments make these measurements by drawing the trajectory traveled by the robots over time, first, tracing their positions on the floor through which the robots navigate, and also tracing their trajectories through the pheromone evaporation simulation. By plotting the robots positioning on the texture used as CA lattice, it is possible to observe how the robots explore the environment, similarly to what was done in the discretized simulations in the Python programming language. Whenever the robots move, a trace of their location on the ground is left, which allows measuring both the movement with the step count and the pheromones evaporation. The images in Fig. 23 that have a more translucent trace are, therefore, the simulations that used an evaporation factor, and those that present an opaque trace, are the ones that mark the time steps covered by the robots. As time progresses, it is noticed that the environment is explored in a uniform way using both GIACA and GSTIACA model. GSTIACA ends up showing a better performance precisely by repeating less movement steps, due to the robots' queues sharing, generating a global mechanism.

Fig. 23(a) shows GIACA model and we have at $t = 2000$ the three robots depositing, two of them are in the same room and only one of them in a different room. Later, **Fig. 23(e)** shows the three robots occupying the same room. Then, **Fig. 23(i)** shows each of the three robots occupies a different room. **Fig. 23(m)** shows 2 robots in the same room and a robot in a room adjacent to the room occupied by the other 2 robots. In time $t \approx 8080$ the GIACA model can complete a task point by spending 6.69×10^4 s. Finally, **Fig. 23(q)** shows two robots practically in adjacent cells and one robot in a room adjacent to the previous one. Thus, in this case, the experiment does not demonstrate good environmental coverage. On the other hand, **Fig. 23(c)** shows a faster evaporation level in the environment and at $t = 2000$ we have two robots in the same room and the other robot finds on the opposite side

in an adjacent room. Fig. 23(g) shows practically the same distribution as the previous scenario. At $t = 6000$, Fig. 23(k) shows a distribution coincidentally similar to the previous scenario. In step $t \approx 7850$ the GIACA model managed to complete a task point, spending approximately 6.34×10^4 s. Fig. 23(o) shows a scenario in which each robot is in a different room. In the last time step represented by Fig. 23(s) in which each robot is in a different room. Thus, it is possible to visually observe that in GSTIACA, although the global information, which is represented by the pheromone deposition, is evaporated and accumulated in smaller quantities in the environment, even so, the robots achieve a very interesting spreading through the environment. The total $T = 10^4$ simulation time for the pheromone map experiment was run for GIACA at 8.26×10^4 s, and GSTIACA the total time was 8.09×10^4 s.

The experiment that shows the coverage quality by the $N = 3$ e-Puck robots in the simulation environment for $T = 10^4$ steps. Fig. 23(b) indicates that 4 rooms were visited and patrolled by robots for GIACA, while for GSTIACA in Fig. 23(d) we have 5 rooms that have already been patrolled in the environment. The same scenario is repeated in Fig. 23(b) and in Fig. 23(h). Fig. 23(j) shows six rooms visited by robots, while Fig. 23(l) shows only five rooms have been patrolled. In step $t \approx 7850$ the GSTIACA model gets the first task point, spending approximately 6.70×10^4 s. Fig. 23(n) and Fig. 23(p) show that all the rooms have already been fully patrolled, but the p_3 robot is further ahead in GSTIACA than in GIACA. In step $t \approx 8080$ the GIACA model managed to complete a task point, spending approximately 7.11×10^4 s. Finally, Fig. 23(r) and Fig. 23(t) show both models have 7 rooms visited, as previously said, however, GSTIACA had a worse environment coverage in relation to GIACA, which still has cells not visited by the robots team. The total $T = 10^4$ simulation time for the Webots environment coverage experiment was run for GIACA at 8.78×10^4 s, and GSTIACA the total time was 8.54×10^4 s.

5. Results discussion

In the simulations carried out with the GIACA and GSTIACA models, the robots were randomly distributed throughout the environment, all of these agents started to perform patrolling in the environment, and a final total time T was given to the team to carry out this task. The environment stores and makes available the information of other robots globally from the sharing of that information and, in addition, there is local information by which each robot makes the best possible decision. It was possible to evaluate that in all simulations GSTIACA overcame GIACA. The main conclusions and results observations of GSTIACA (the best model) are: (a) an adequate adjustment using GA meta-optimization of the parameters involved was important to obtain the best robot team performance, and $\{P_{cross} = 0.6, P_{mu} = 0.1, P_{size} = 300, T_{cross} = 180, T_{offspring} = 90\}$; (b) the robot team's ability to share information (globally – pheromone trails where $\tau_{max} = 30.364$ - and locally – when robots are in the same room Ω_i they check other tabu queue $\overline{CQ} = 1$) accelerates the team's efficiency in completing the task; however, the model can be applied even if these skills are not available (and also including $QP_\otimes = 0$ and $QP_\emptyset = 0$, for queues policies in deadlock and full case, respectively); (c) short-term memory in the search also accelerates the team's efficiency, and this improvement is even more noticeable when the other constants are optimized by GA; however, excessive usage of this resource takes expensive processing time, making intermediate memory sizes more appropriate (in this work, $|Q| = 14$); (d) an excessive robots number N in the team causes a significant improvement in the team's performance, depending on the rooms number to be visited, however, this increase can generate congestion and bottleneck in the passage through the doors; (e) some observations related to the environment L characteristics, such as the position and obstacles' complexity, as well as the environment size and configuration, can impair the team's performance due to the high processing time, here the environment complexity is $E_1 < E_2 < E_4 < E_3$; (f) queue sharing prevents two or more robots from remaining in the same room (Ω_i), reducing the cycles number spent until all rooms are properly patrolled, however, the queuing policies optimized by GA were fundamental to deal with deadlock $QP_\otimes = 1$ or full memories cases $QP_\emptyset = 1$; (g) based on the pheromones repulsive effect (global effect) and the memory (local and global effect) usage to avoid recently visited regions, the computational model returns a good environment coverage; here $\delta = 0.859$ and the spread force $\sigma = 0.016$; (h) an adequate memory size specification is relevant for the coverage: small or nonexistent memories lead to a low visited rooms and many overloaded regions, while large memories increase the processing iteration time, because the robot access can be obstructed, generating deadlocks in specific environments corners; (i) the improvement in the artificial pheromone decline (here $\beta = 0.322$) mimics the chemical ants pheromone; (j) the usage of CA asynchronous pheromone update time of decline process provides an improvement in the processing times of both the GIACA and GSTIACA models investigated here making them about 24 times faster (considering the same environments configurations) than the IACA models [69] and TIACA [75] investigated in precursor works; (k) the robot's next cell selection provides a first choice selection [4,30,36] and it is based in a greedy randomized adaptative search procedure [82]; (l) the combination of different natural computing techniques, such as cellular automata, genetic algorithms, inverted pheromone from ant colonies and Tabu Search, greedy randomized search, as well physical forces that mimics pedestrian evacuation behavior led the surrogate model creation to be implemented in a real world robotic architecture, as demonstrated by experiments on Webots using the e-Puck robot.

6. Conclusions and future work

This work proposes and investigates two new control models for swarm robotics based on two-dimensional cellular automata, here called GIACA and GSTIACA. Overall, the robot movement rules in our models were based on the pedestrian social interaction rules of the works [30,31]. Additionally, ants' inverted pheromone refers to the fact that when these insects are at risk, they tend to drift away from each other, which we used in our work to increase the coverage area. Unlike when ants are in a foraging situation, they tend to cluster along specific paths that lead to the food source [41]. Furthermore, in this work we use direct and indirect communication mechanisms between robotic agents, similar to the behavior of ants [41]. In global indirect communication, the pheromone is spread through the lattice. While in direct communication, a queue based on Tabu Search is used individually and



Fig. 23. Step and pheromone maps for GIACA and GSTIACA using an environment modification E_1 with 60×80 cells, where p_1 is represented by (■), p_2 is represented by (●) and p_3 is represented by (▲) during $T = 10^4$ time steps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

shared, to prevent robots from visiting places recently visited by the team. Further reinforcing the repulsion among agents to increase the coverage of the environment and consequently improve the quality of the patrol. The union of these social computing methods has given rise to a new bio-inspired model for controlling swarms of robots. This means that, these models were integrated in a CA controller that served as a surrogate for the real-robots simulation. Due to the large number of parameters found after the unification of all models, these parameters were optimized by an evolutionary computation technique called genetic algorithm.

A surrogate model is an engineering approach that is used when an outcome of interest cannot be easily directly measured, in this case, due to the size of the environment to be simulated, which was $280 \text{ cm} \times 420 \text{ cm}$, in software simulation or $560 \text{ cm} \times 840 \text{ cm}$, in Webots simulator using e-Puck architecture, with a robot size of only 7 cm, a simulation of more than a week would be necessary, so we used the merged methodology between CA and GA for the creation of a surrogate model that presents the controller created herein, without the need to do the simulation with real-robots. Our model is dedicated to the surveillance task, which is very relevant to collective robotics: (i) it is a metaphor for a broad class of problems that integrate exploration, navigation and objects transportation; (ii) surveillance is a canonical problem for the study of robot–robot cooperation in systems with several robots; (iii) many real-world applications for robotics, mining, cleaning, rescue and exploration are robot patrolling instances.

Three main processes were investigated in this work, and, in the first process, we improved the CA-based robot control model for surveillance, which in this case was the adoption of queue sharing based on the Tabu search mechanism. The second is the usage of CA asynchronous pheromone decline rule update. In the last one, we used genetic algorithms for the meta-optimization of surrogate parameters for IACA and STIACA surveillance, and with their optimized parameters were called GIACA and GSTIACA, respectively. As the system was based on a biological system, we can say that it is robust. There is a guarantee that even if one or more characteristics change (system disturbances), the robot examination is still able to perform the task patrolling the environment.

The GIACA research is guided by the IACA [69] parameters meta-optimization, by the repulsive pheromone [6] – distributed by robots throughout the environment while they walk on it – and by a short-term individual memory based on the Tabu search algorithm [47,48]. The GIACA meta-optimization is based on natural computing concepts for the IACA parameters evolution, thus resulting in a optimized model through the GA application, which we call GIACA. In addition, this model is also inspired by previous CA-based crowd dynamics models during pedestrian evacuation [30,31,36,85] and robot navigation models for the foraging task as in CAAM [69], PCAAM [66] or SCAAM [17]. Each robot chooses an ideal route to follow and travels through a room, considering only the global pheromones grid. The robots use a greedy randomized process to select next cell movement [82], called here first choice movement [4,30,31] GSTIACA in turn is a model based on TIACA [75], an IACA improvement, however in that model the use of swarm robotics was not adopted, that is, only one robot was used in the surveillance task. However, in STIACA, the queues sharing between robots within the same environment made the model achieve a superior performance in relation to the precursors. In addition, a strategy to implement the pheromone decline update has made our model faster (in computational time) than its precursors, since the lattice is updated only when it needs to be used, also called lazy evaluation method. Finally, the adoption of evolutionary computation through genetic algorithms for the interaction and evolution of all these parameters at the same time, made GSTIACA superior to its precursors, IACA [69], TIACA [75], and even GIACA, also proposed here.

Additionally, we used the controlling foraging agents idea and pedestrians in a panic evacuation situation [30,31,83] based on CA to create our new GSTIACA navigation model for swarm robotics. It is important to emphasize that the objective of these works was to reproduce the human behavior observed in specific situations [30,31], and it is not the objective of these studies to propose ways to change this behavior, as it is basically instinctive. The objective was to carry out a simulation as close as possible to the observed phenomenon and then seek to understand the environmental factors that could affect this behavior, in order to propose improvements in the environment that would facilitate evacuation in emergency situations. In the models for the robotic foraging task CAAM [4], PCAAM [66] and SCAAM [17], the objective was to create a new model based on CA that would mimic this human behavior in swarm robotics. In the same way, this idea was used in the precursor works that investigated the robotic surveillance task, such as in IACA [69] and TIACA [94]. In the present work, these previous models of natural human behavior were adapted to provide efficient behavior for the robot team when searching for the nearest collection point. As far as we know, no previous work has drawn this parallel between the use of different natural computing techniques for the robots control, such as ant colonies, Tabu search, genetic algorithms, cellular automata. Additionally, we also made an adaptation in the pheromone decline updating function. In this case we used an asynchronous CA update to optimize the GA processing time, which until then had not been mentioned by any other author, in this specific context.

Other robotics works also used the ant pheromone propagation metaphor to control the robots trajectories [64]. However, the usual pheromone usage is similar to that used by ants in nature: creating an attraction effect that makes robots behave as followers. In modeling the surveillance process in the GSTIACA model, most of the inspirations were the precursor model proposed in [6,42,99], which uses the idea of propagating an inverted pheromone applied in the surveillance task IACA [69], TIACA [75], and CAAM [4] which has in common with the task investigated here the need for good environment coverage. This coverage is obtained through the repulsive effect between robots, which leads to a natural swarm robots spread over environment. However, the novelty used in the present work is the fact that this dynamic has been reinforced by the use of a short-term memory that is now shared between robots within the same room, which also prevents the robots from returning to recently visited cells, forcing them to patrol new areas that the team has not yet visited. Another difference between the present work and the model in [6,42,99] is that the modeling here was completely discrete for both the pheromone deposit and the movement stage, allowing local conflicts control by the transition rules while in the previous work the authors mention the use of an obstacle prevention algorithm to avoid collisions with the environment and other robots.

The simulations were performed with the GSTIACA model, using different surveillance scenarios, for a good parameters adjustment and a better understanding of the model's behavior. The main experiments conclusions were summarized in the last section. The proposed model implementation made it possible to generate visual data on the behavior patterns observed in ant colonies and even on the pedestrians dynamics or swarm robots. In addition, statistical data were obtained relating the task points with the robots amount allocated in the simulation environment. These data were collected and studied statistically; initiated an analysis that can be deepened, leading to subsequent research.

We also implemented our GSTIACA model in the Webots simulator using the e-Puck [67] robotic architecture. The results were compared with GIACA and we also obtained superior results for GSTIACA. The advantage of performing simulations on this platform is that they take into account architectures and environments physical aspects, allowing a more realistic analysis of the model's performance using robotic architectures from the real world. Relevant issues, such as the need for synchronization [67] and information accuracy dependence, were better evaluated in these simulations.

As a continuation of this work, we believe that we can implement the model with real robotic architectures. Another continuity that has already begun is the model adaptation to other robotic tasks that have new peculiarities to be addressed, such as the selective garbage collection [17], the foraging [66] and the search and rescue [100]. We also intend to use other metaheuristics algorithms to improve and refine our model and compare the results found with the model presented here and other precursor models. In addition, we believe that the fact that the computational model is managed by simple transition rules will allow us to add new parameters for different social or physical behaviors, which can improve the team robots' performance.

7. Symbols list

Below are explained the symbols that were used throughout the text.

N	number of robots
T	total number of iterations
x_{ij}	current CA lattice cells
t	time steps
r	CA radius
m	the number of time steps that the cell remained without updating
v	CA Moore's neighborhood
L	CA lattice, floor field or grid
η_{ij}^v	CA Moore's neighborhood or vicinity
τ_{max}	maximum pheromone per cell
δ	pheromone deposition value
σ	pheromone dispersion value
β	declining pheromone value
d	parents selected by the selection criteria
d'	descendants of parents (d_1, d_2)
α	mask for GA crossover
T_p	population size
$T_{offspring}$	tamanho da offspring
P_{mut}	percentage of individuals who will mutate
T_{mut}	number of individuals that will mutate
P_{cross}	percentage of individuals selected for crossover
T_{cross}	number of individuals selected for crossover
\overline{CQ}	check other tabu queue
QP_{\otimes}	tabu query policy deadlock case
QP_{\emptyset}	tabu query policy full case
$ Q $	tabu size
CT	completed tasks
TP	task points
p	robot identifier
E	environment identifier
SM	stepmap values
Ph	pheromone map values
Ω	distance for direct communication
v_{norm}	normalization value GA
v_i	possible values GA
\bar{x}	mean
S	standard deviation S of population
k	amount of times the GA is executed to calculate the fitness
F_d	GA fitness
L	CA grid or lattice
y_{CT}	completed tasks:y axis
x_N	quantity of robots used: x axis
t_s	task start:initial time that the first task was completed
\bar{x}	mean used to represent GA fitness
t_f	task finish:final moment that the last task was completed
\bar{T}	upper whiskers
\perp	lower whiskers
\circ	outliers
\square	quartiles (y-axis)
SM_b	steps map color minimum value (blue)
SM_r	steps map color maximum value (red)
Ph_b	pheromones map color minimum value (blue)
Ph_r	pheromones map color maximum value (red)
$x_{(i+a)(j+b)}$	CA vicinity cells around current cell

⊢ perpendicular collisions in corners
 | represents each task point
 x_N regression quantity of robots
 y_{CT} regression amount of tasks completed
 G quantity of generations

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors DAL and HJML are grateful to National Council for Scientific and Technological Development (CNPq), Brazil, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil and Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), Brazil for financial support. DAL is grateful to Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PROPI) and Núcleo de Inovação Tecnológica (NIT) of Instituto Federal do Triângulo Mineiro (IFTM) for funding.

References

- [1] Mendonça M, Chrun IR, Neves Jr. F, Arruda LV. A cooperative architecture for swarm robotic based on dynamic fuzzy cognitive maps. *Eng Appl Artif Intell* 2017;59:122–32.
- [2] Suárez P, Iglesias A, Gálvez A. Make robots be bats: specializing robotic swarms to the bat algorithm. *Swarm Evol Comput* 2019;44:113–29.
- [3] Bakhshipour M, Ghadi MJ, Namdari F. Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach. *Appl Soft Comput* 2017;57:708–26.
- [4] Lima DA, Oliveira GM. A cellular automata ant memory model of foraging in a swarm of robots. *Appl Math Model* 2017;47:551–72.
- [5] Inácio FR, Macharet DG, Chaimowicz L. PSO-based strategy for the segregation of heterogeneous robotic swarms. *J Comput Sci* 2019;31:86–94.
- [6] Calvo R, de Oliveira JR, Figueiredo M, Romero RAF. Bio-inspired coordination of multiple robots systems and stigmergy mechanisms to cooperative exploration and surveillance tasks. In: *Cybernetics and intelligent systems*. IEEE; 2011, p. 223–8.
- [7] Castello E, Yamamoto T, Dalla Libera F, Liu W, Winfield AF, Nakamura Y, et al. Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach. *Swarm Intell* 2016;1:1–31.
- [8] Costa Filho RN, Paucar VL. A multi-objective optimization model for robust tuning of wide-area PSSs for enhancement and control of power system angular stability. *Results Control Optim* 2021;3:100011.
- [9] Hasan S, Dhingra AK. Performance verification of different control schemes in human lower extremity rehabilitation robot. *Results Control Optim* 2021;100028.
- [10] Lima DA, Ferreira MEA, Silva AFF. Machine learning and data visualization to evaluate a robotics and programming project targeted for women. *J Intell Robot Syst* 2021;103(4):1–20.
- [11] Jayne D, Pigazzi M, Marshall H, Croft J, Corrigan N, Copeland J, et al. Robotic-assisted surgery compared with laparoscopic resection surgery for rectal cancer: the ROLARR RCT. 2019.
- [12] Stefanec M, Szopek M, Schmickl T, Mills R. Governing the swarm: Controlling a bio-hybrid society of bees & robots with computational feedback loops. In: *2017 IEEE symposium series on computational intelligence*. IEEE; 2017, p. 1–8.
- [13] Allwright M, Bhalla N, Pincioli C, Dorigo M. Simulating multi-robot construction in ARGoS. In: *International conference on swarm intelligence*. Springer; 2018, p. 188–200.
- [14] Sun X, Liu T, Hu C, Fu Q, Yue S. Colcos φ : A multiple pheromone communication system for swarm robotics and social insects research. In: *2019 IEEE 4th international conference on advanced robotics and mechatronics*. IEEE; 2019, p. 59–66.
- [15] Verlekar H, Joshi K. Ant & bee inspired foraging swarm robots using computer vision. In: *2017 International conference on electrical, electronics, communication, computer, and optimization techniques*. IEEE; 2017, p. 191–5.
- [16] Xue S, Claudio F, Shi X, Li T. Revealing the hidden rules of bidirectional pedestrian flow based on an improved floor field cellular automata model. *Simul Model Pract Theory* 2020;100:102044.
- [17] Lima DA, Oliveira GM. Stochastic cellular automata ant memory model for swarm robots performing efficiently the garbage collection task. In: *2019 19th International conference on advanced robotics*. IEEE; 2019, p. 708–13.
- [18] Alitappeh RJ, Jeddisaravi K. Multi-robot exploration in task allocation problem. *Appl Intell* 2022;52(2):2189–211.
- [19] Oliveira GM, Vargas PA, Ferreira GB. A local decision making cellular automata-based path-planning. In: *11th National meeting on artificial and computational intelligence*. 2014, p. 1–6.
- [20] Lopes HJM, Lima DA. Cellular automata in path planning navigation control applied in surveillance task using the e-puck architecture. In: *2020 IEEE international conference on systems, man, and cybernetics*. 2020, p. 1117–22.
- [21] Liu Y, Li H, Liu Z. Real-time coverage path planning of a UAV with threat and value zone constraints. In: *Chinese intelligent systems conference*. Springer; 2020, p. 840–8.
- [22] Corke P, Roberts J, Winstanley G. Sensors and control for mining robotics. In: *Proceedings of the fourth international symposium on mine mechanisation and automation*. Cooperative Research Centre for Mining Technology and Equipment; 1997, p. B1–11.
- [23] Koes M, Nourbakhsh I, Sycara K. Constraint optimization coordination architecture for search and rescue robotics. In: *Proceedings 2006 IEEE international conference on robotics and automation*, 2006. IEEE; 2006, p. 3977–82.
- [24] Ahangaran M, Taghizadeh N, Beigy H. Associative cellular learning automata and its applications. *Appl Soft Comput* 2017;53:1–18.
- [25] Kotyrba M, Volna E, Bujok P. Unconventional modelling of complex system via cellular automata and differential evolution. *Swarm Evol Comput* 2015;25:52–62.
- [26] Lima HA, Lima DA. Automatos celulares estocasticos bidimensionais aplicados a simulacao de propagacao de incendios em florestas homogeneas. In: *Workshop of applied computing for the management of the environment and natural resources (WCAMA) congresso da sociedade brasileira de computacao (2014 CSBC)*. 2014.
- [27] Monteiro L, Fanti V, Tessaro A. On the spread of SARS-CoV-2 under quarantine: A study based on probabilistic cellular automaton. *Ecol Complex* 2020;44:100879.

- [28] Seck-Tuoh-Mora JC, Hernandez-Romero N, Lagos-Eulogio P, Medina-Marin J, Zuñiga-Peña NS. A continuous-state cellular automata algorithm for global optimization. *Expert Syst Appl* 2021;177:114930.
- [29] Lima MVB, Oliveira CC, Lima DA. Uma ferramenta computacional para simulacao de espalhamento de fluidos baseada em automatos celulares bidimensionais estocasticos. In: Workshop of applied computing for the management of the environment and natural resources. 2016.
- [30] Varas A, Cornejo M, Mainemer D, Toledo B, Rogan J, Munoz V, et al. Cellular automaton model for evacuation process with obstacles. *Physica A* 2007;382(2):631–42.
- [31] Alizadeh R. A dynamic cellular automaton model for evacuation process with obstacles. *Saf Sci* 2011.
- [32] Silva EC, Soares JA, Lima DA. Autómatos celulares unidimensionais caóticos com borda fixa aplicados à modelagem de um sistema criptográfico para imagens digitais. *Rev Inform Teórica E Aplicada* 2016;23(1):250–76.
- [33] Lira ER, de Macêdo HB, Lima DA, Alt L, Oliveira G. A reversible system based on hybrid toggle radius-4 cellular automata and its application as a block cipher. 2021, arXiv preprint [arXiv:2106.04777](https://arxiv.org/abs/2106.04777).
- [34] Seck-Tuoh-Mora JC, Hernandez-Romero N, Medina-Marin J, Martinez GJ, Barragan-Vite I. Random expansion method for the generation of complex cellular automata. *Inform Sci* 2021;549:310–27.
- [35] Tian N, Sun J, Wang M, Qiu L. Artificial bee colony with cellular automata. *ICIC Express Lett* 2018;12(2):125–34.
- [36] Lima DA, Cabral Jr. E, Almeida IT, Andrade JP, Fonseca JP, Santos ME, et al. A fire elitist cellular automaton-based model to verify pedestrian flow simulated in real environments using arduino. *Proc Ser Braz Soc Comput Appl Math* 2020;7(1).
- [37] Zuñiga-Peña NS, Hernández-Romero N, Seck-Tuoh-Mora JC, Medina-Marin J, Barragan-Vite I. Improving 3D path tracking of unmanned aerial vehicles through optimization of compensated PD and PID controllers. *Appl Sci* 2021;12(1):99.
- [38] Vargas PA, Di Paolo EA, Harvey I, Husbands P. The horizons of evolutionary robotics. MIT Press; 2014.
- [39] Wolfram S. A new kind of science. Wolfram Media; 2002.
- [40] Dorigo M, Birattari M, Blum C, Clerc M, Stützle T, Winfield A. Ant colony optimization and swarm intelligence. In: 6th International conference, ants 2008, Brussels, Belgium, september 22–24, 2008, proceedings, Vol. 5217. Springer; 2008.
- [41] Gordon DM. The ecology of collective behavior. *PLoS Biol* 2014;12(3):e1001805.
- [42] Calvo R, de Oliveira JR, Figueiredo M, Romero RA. Parametric investigation of a distributed strategy for multiple agents systems applied to cooperative tasks. In: Proceedings of the 29th annual ACM symposium on applied computing. ACM; 2014, p. 207–12.
- [43] Ioannidis K, Sirakoulis GC, Andreadis I. Cellular automata-based architecture for cooperative miniature robots.. *J Cell Autom* 2013;8.
- [44] Lima DA, Oliveira GM. Formal analysis in a cellular automata ant model using swarm intelligence in robotics foraging task. In: 2017 IEEE international conference on systems, man, and cybernetics. IEEE; 2017, p. 1793–8.
- [45] Zou J, Zhao Q, Yang W, Wang F. Occupancy detection in the office by analyzing surveillance videos and its application to building energy conservation. *Energy Build* 2017;152:385–98.
- [46] Zhang C, Jia Q-S. An occupancy distribution estimation method using the surveillance cameras in buildings. In: 2017 13th IEEE conference on automation science and engineering. IEEE; 2017, p. 894–9.
- [47] Glover F. Tabu search Part I. *ORSA J Comput* 1989;1(3):190–206.
- [48] Glover F. Tabu search Part II. *ORSA J Comput* 1990;2(1):4–32.
- [49] Tinoco CR, Lima DA, Oliveira GM. An improved model for swarm robotics in surveillance based on cellular automata and repulsive pheromone with discrete diffusion. *Int J Parallel Emergent Distrib Syst* 2017;34(1):53–77.
- [50] Liu W, Winfield AF, Sa J, Chen J, Dou L. Towards energy optimization: Emergent task allocation in a swarm of foraging robots. *Adapt Behav* 2007;15(3):289–305.
- [51] Sun J, Xu W, Feng B. A global search strategy of quantum-behaved particle swarm optimization. In: IEEE conference on cybernetics and intelligent systems, 2004, Vol. 1. IEEE; 2004, p. 111–6.
- [52] Selman B, Kautz HA, Cohen B. Noise strategies for improving local search. In: AAAI, Vol. 94. 1994, p. 337–43.
- [53] Powathil GG, Gordon KE, Hill LA, Chaplain MAJ. Modelling the effects of cell-cycle heterogeneity on the response of a solid tumour to chemotherapy: Biological insights from a hybrid multiscale cellular automaton model. *J Theoret Biol* 2012;308:1–19.
- [54] Mitchell M, Crutchfield JP, Das R, et al. Evolving cellular automata with genetic algorithms: A review of recent work. In: Proceedings of the first international conference on evolutionary computation and its applications. Moscow; 1996.
- [55] Grefenstette JJ. Optimization of control parameters for genetic algorithms. *IEEE Trans Syst Man Cybern* 1986;SMC-16(1):122–8.
- [56] Bohaienko V. Selection of ψ -Caputo derivatives' functional parameters in generalized water transport equation by genetic programming technique. *Results Control Optim* 2021;100068.
- [57] Dong H, Li T, Ding R, Sun J. A novel hybrid genetic algorithm with granular information for feature selection and optimization. *Appl Soft Comput* 2018;65:33–46.
- [58] Pohlheim H. Genetic and evolutionary algorithm toolbox for use with MATLAB. Dept Comput Sci, Univ Ilmenau, Ilmenau, Germany 1998. Retrieved from http://www.geatbx.com/download/geatbx_tutorial_v33c.pdf.
- [59] Jin Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm Evol Comput* 2011;1(2):61–70.
- [60] Yan X, Abdel-Aty M, Radwan E, Wang X, Chilakapati P. Validating a driving simulator using surrogate safety measures. *Accid Anal Prev* 2008;40(1):274–88.
- [61] Liu Y, Collette M. Improving surrogate-assisted variable fidelity multi-objective optimization using a clustering algorithm. *Appl Soft Comput* 2014;24:482–93.
- [62] Sun X, Gong D, Jin Y, Chen S. A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning. *IEEE Trans Cybern* 2013;43(2):685–98.
- [63] Pehlivanoglu YV, Yagiz B. Aerodynamic design prediction using surrogate-based modeling in genetic algorithm architecture. *Aerosp Sci Technol* 2012;23(1):479–91.
- [64] Ioannidis K, Sirakoulis GC, Andreadis I. A path planning method based on cellular automata for cooperative robots. *Appl Artif Intell* 2011;25(8):721–45.
- [65] Nishinari K, Sugawara K, Kazama T, Schadschneider A, Chowdhury D. Modelling of self-driven particles: Foraging ants and pedestrians. *Physica A* 2006;372(1):132–41.
- [66] Lima DA, Oliveira GMB. A probabilistic cellular automata ant memory model for a swarm of foraging robots. In: Control, automation, robotics and vision, 2016 (ICARCV). 14th International conference on, Vol. 1. IEEE; 2016, p. 1–6.
- [67] Lima DA, Tinoco CR, Viedman JM, Oliveira GM. Coordination, synchronization and localization investigations in a parallel intelligent robot cellular automata model that performs foraging task. In: ICART (2). 2017, p. 355–63.
- [68] Lima DA, Oliveira GM. New bio-inspired coordination strategies for multi-agent systems applied to foraging tasks. In: 2016 IEEE 28th international conference on tools with artificial intelligence. IEEE; 2016, p. 1–8.
- [69] Lima DA, Tinoco CR, Oliveira GMB. A cellular automata model with repulsive pheromone for swarm robotics in surveillance. In: Cellular automata - international conference on cellular automata for research and industry, ACRI. Proceedings. 2016, p. 312–22.
- [70] Holland JH. Genetic algorithms and adaptation. In: Adaptive control of ill-defined systems. Springer; 1984, p. 317–33.
- [71] Tinoco CR, Oliveira GM. Pheromone interactions in a cellular automata-based model for surveillance robots. In: International conference on cellular automata. Springer; 2018, p. 154–65.

- [72] Tinoco CR, Oliveira GM. Heterogeneous teams of robots using a coordinating model for surveillance task based on cellular automata and repulsive pheromone. In: 2019 IEEE congress on evolutionary computation. IEEE; 2019, p. 747–54.
- [73] Tinoco C, Vizzari G, Oliveira G. Parameter adjustment of a bio-inspired coordination model for swarm robotics using evolutionary optimisation. In: Lecture notes in computer science. 2021, p. 146–55. http://dx.doi.org/10.1007/978-3-030-69480-7_15.
- [74] Lopes HJ, Lima DA. Evolutionary tabu inverted ant cellular automata with elitist inertia for swarm robotics as surrogate method in surveillance task using e-puck architecture. *Robot Auton Syst* 2021;103840.
- [75] Souza NLB, Lima DA. Tabu search for the surveillance task optimization of a robot controlled by two-dimensional stochastic cellular automata ants model. In: 2019 Latin American robotics symposium (LARS), 2019 Brazilian symposium on robotics (SBR) and 2019 workshop on robotics in education. IEEE; 2019, p. 299–304.
- [76] Lopes HJ, Lima DA. Patrolling simulation model for swarm robotics using ant memory cellular automata maps with genetic algorithms optimization. *Expert Syst Appl* 2021. in press.
- [77] Lima, et al. A cellular automata model with repulsive pheromone for swarm robotics in surveillance. In: International Conference on Cellular Automata. Springer; 2016, p. 312–22.
- [78] Slimi R, El Yacoubi S, Dumonteil E, Gourbiere S. A cellular automata model for chagas disease. *Appl Math Model* 2009;33(2):1072–85.
- [79] Silva EC, Soares JAJP, Lima DA. One-dimensional chaotic cellular automata with fixed border applied to a cryptosystem modeling for digital images. *Rev Inform Teorica E Aplicada* 2016;23:250–76.
- [80] Kumar N, Mahato SK, Bhunia AK. Design of an efficient hybridized CS-PSO algorithm and its applications for solving constrained and bound constrained structural engineering design problems. *Results Control Optim* 2021;100064.
- [81] Bandyopadhyay R, Basu A, Cuevas E, Sarkar R. Harris hawks optimisation with simulated annealing as a deep feature selection method for screening of COVID-19 CT-scans. *Appl Soft Comput* 2021;111:107698.
- [82] Resende MG, Ribeiro CC. Greedy randomized adaptive search procedures. In: Handbook of metaheuristics. Springer; 2003, p. 219–49.
- [83] Schadschneider A, Klingsch W, Klüpfel H, Kretz T, Rogsch C, Seyfried A. Evacuation dynamics: Empirical results, modeling and applications. In: Extreme environmental events. Springer; 2011, p. 517–50.
- [84] Mondada F, Bonani M, Raemy X, Pugh J, Cianci C, Klaptoz A, et al. The e-puck, a robot designed for education in engineering. In: Proceedings of the 9th conference on autonomous robot systems and competitions, Vol. 1. IPCB: Instituto Politécnico de Castelo Branco; 2009, p. 59–65.
- [85] Perez GJ, Tapang G, Lim M, Saloma C. Streaming, disruptive interference and power-law behavior in the exit dynamics of confined pedestrians. *Physica A* 2002;312(3–4):609–18.
- [86] Winfield AF. Foraging robots. In: Encyclopedia of complexity and systems science. Springer; 2009, p. 3682–700.
- [87] Del Ser J, Osaba E, Molina D, Yang X-S, Salcedo-Sanz S, Camacho D, et al. Bio-inspired computation: Where we stand and what's next. *Swarm Evol Comput* 2019;48:220–50.
- [88] Calvo R, et al. Bio-inspired coordination of multiple robots systems and stigmergy mechanisms to cooperative exploration and surveillance tasks. In: 2011 IEEE 5th international conference on cybernetics and intelligent systems. 2011, p. 223–8.
- [89] Han J, Pei J, Kamber M. Data mining: Concepts and techniques. Elsevier; 2011, p. 114.
- [90] Sivanandam S, Deepa S. Introduction to genetic algorithms. Springer Science & Business Media; 2007.
- [91] Shu LH, Flowers WC. Towards life-cycle fastening and joining cost optimization using genetic algorithms. 1996.
- [92] Rogers A, Prugel-Bennett A. Genetic drift in genetic algorithm selection schemes. *IEEE Trans Evol Comput* 1999;3(4):298–303.
- [93] Pillai AC, Thies PR, Johanning L. Mooring system design optimization using a surrogate assisted multi-objective genetic algorithm. *Eng Optim* 2018.
- [94] de Souza JR, Pessin G, Osório FS, Wolf DF, Vargas PA. Combining evolution and training in a robotic controller for autonomous vehicle navigation. In: Advances in autonomous robotics. Springer; 2012, p. 426–7.
- [95] Michel O. Cyberbotics Ltd. Webots: professional mobile robot simulation. *Int J Adv Robot Syst* 2004;1(1):5.
- [96] Marchese FM. Time-invariant motion planner in discretized C-spacetime for MRS. *Multi-Robot Syst Trends Dev* 2011;307–24.
- [97] Martinelli A. The odometry error of a mobile robot with a synchronous drive system. *Robotics Autom, IEEE Trans* 2002;18(3):399–405.
- [98] Mondada F, Bonani M, Raemy X, Pugh J, Cianci C, Klaptoz A, et al. The e-puck, a robot designed for education in engineering. In: Proceedings of the 9th conference on autonomous robot systems and competitions, Vol. 1. IPCB: Instituto Politécnico de Castelo Branco; 2009, p. 59–65.
- [99] Calvo R, de Oliveira JR, Romero RA, Figueiredo M. A bioinspired coordination strategy for controlling of multiple robots in surveillance tasks. *Int J Adv Softw* 2012;5(3 & 4).
- [100] Kantor G, Singh S, Peterson R, Rus D, Das A, Kumar V, et al. Distributed search and rescue with robot and sensor teams. In: Field and service robotics. Springer; 2006, p. 529–38.