Formalisation and validation through Coloured Petri Nets of a bio-inspired coordination model for swarms of robots

Claudiney R. Tinoco
claudineyrt@gmail.com
Federal University of Uberlândia - UFU
Uberlândia, MG, Brazil

Leiliane P. Rezende
rezende@utfpr.edu.br
Federal Technological University of Paraná - UTFPR
Santa Helena, PA, Brazil

Gina M. B. Oliveira
gina@ufu.br
Federal University of Uberlândia - UFU
Uberlândia, MG, Brazil

Stéphane Julia
stephane@ufu.br
Federal University of Uberlândia - UFU
Uberlândia, MG, Brazil

ABSTRACT

Multi-robots and swarms of robots systems, especially those that have bio-inspired techniques, lack a formal method for modelling and analysis. The problem lies in the fact that bio-inspired algorithms are intrinsically hard to be analysed due to their evolutionary nature, and this difficulty increases when these algorithms are applied to swarms. In this work, the application of Coloured Petri Nets is proposed to carry out the formal validation of the coordination model for swarm of robots IACA-DI, which was proposed for surveillance, coverage, foraging and search-and-rescue tasks. According to the outcomes, it was possible to demonstrate that the basic requirements for a swarm robotics coordination model were achieved. This is due to the fact that the Petri net has proven that the system is very reliable, has no deadlocks and has the restartability property, which is very important for the proposed task.

CCS CONCEPTS

• **Software and its engineering** → **Formal software verification**; **Process validation**; • **Computing methodologies** → **Simulation tools**; **Simulation evaluation**.

KEYWORDS

Swarm robotics, Coloured Petri Nets, Modelling, Formalisation, WorkFlow Nets, State Graphs, CPN Tools

ACM Reference Format:
Claudiney R. Tinoco, Leiliane P. Rezende, Gina M. B. Oliveira, and Stéphane Julia. 2018. Formalisation and validation through Coloured Petri Nets of a bio-inspired coordination model for swarms of robots. In *Proceedings of Simpósio Brasileiro de Engenharia de Software (SBES).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

SBES, Out 03–07, 2022, Uberlândia, MG

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The mimicry of the behaviour of social animals inspires the development and organisation of swarms of robots. In most research, these animals are social insects, performing some type of cooperative task (e.g., foraging) [9]. The local and indirect interactions with other insects allow the execution of complex tasks using just simple local interactions, i.e., through local interactions emerges a complex global behaviour. To be characterised as a robot swarm coordination model, some fundamental principles need to be addressed [5]: (i) the robots are independent; (ii) the robots have some form of indirect communication (or similar), and there is no direct contact with other robots; (iii) the robots have only local interaction forms; (iv) there is no centralised system to control the swarm or some form of global knowledge; and (v) there is a form of cooperation between robots to achieve an objective.

Cooperative behaviour techniques for robotics have been proposed from different inspirations and metaphors, being that bio-inspired strategies represent a large part of the research in the related literature. For instance, ant pheromone-inspired models [3], reproduction of physical phenomena, such as potential fields [26] and fluid dynamics [40], and cellular automata based models for navigation under formation control [13, 19], crowd evacuation [4], path planning [27] and surveillance [34].

In turn, Coloured Petri Nets (CPN) [21] represents processes in dynamic systems in which there are concurrency, parallelism and information synchronisation. The mathematical formalism is one of the most important features of Petri nets, since it allows to carry out an accurate analysis of the models in order to verify structural and behavioural properties. In the related literature, Petri nets have been applied, for example, in the modelling and analysis of video-games [1], formal verification of discrete systems [6, 18], performance appraisal [16], real-time and distributed software design [23, 28], information systems modelling [14], communication protocols [20] and database management [10, 25]. There are also several applications of Petri nets in robotics: task coordination [7, 11], path planning [12, 31, 37], reliability analysis [33] and in the coordination of multi-robots [24, 30].

The lack of a formal method capable of evaluating and proving the effectiveness and efficiency of a model is one of the major gaps found in robotics research, especially when it comes to robotics with multiple robots and swarms of robots. Therefore, in this work, the swarm robotics coordination model *Inverted Ant*

Cellular Automata with Discrete pheromone diffusion and Inertial motion (IACA-DI), proposed in work [34], is modelled and analysed through CPNs. Considering that Petri nets are formal models, it is expected, through the formalisation of the IACA-DI model, that some conclusions of previous works, which were drawn through empirical analysis, can be substantiated. In addition, simulations in a CPN simulation platform (CPN Tools [22]), will make it possible to highlight important features of the model. Features related, for example, to the presence of deadlocks and indispensable properties (e.g., restartability, conservativity and limitability), which must be in accordance with the requirements of the model.

This paper is organised as follows. Section II details the robotic coordination model IACA-DI and the theoretical basis behind the Coloured Petri Nets. The modelling of the IACA-DI model through CPNs is made in Section III. In Section IV, the properties of the modelled CPN are validated by creating the state space graph. Finally, Section V presents the main conclusions and future works.

2 THEORETICAL FRAMEWORK

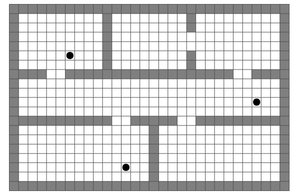
This section presents the Theoretical Framework used in this work, in which it is used in the creation of the proposed CPN. Section 2.1 describes the IACA-DI model and Section 2.2 presents the formalisation of the Coloured Petri Nets.

2.1 IACA-DI: Swarm Robotics Coordination

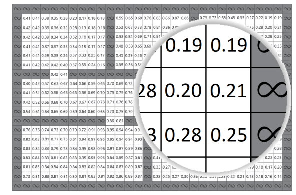
The coordination model for swarms robotics modelled in this work with CPNs is denominated IACA-DI [34]. This model combines bio-inspired strategies to coordinate swarms of robots, and can be applied to tasks like surveillance and exploration. These bio-inspired strategies are Cellular Automata (CAs) [2] and Inverted Ant System (IAS) [8].

Succinctly, CA are applied in the discretization of the environment in a grid of identical squared cells. Then, this grid is duplicated to describe two types of information: (i) a physical grid (Fig. 1a), where states represent the physical elements of the environment (R - Robot, O - Obstacle and F - Free cells); and (ii) a pheromone grid (Fig. 1b), which is used to represent the concentration of pheromone deposited by the robots in each cell of the environment. The pheromone grid has continuous states, which are defined by values between [0.0, 1.0]. Whereas the robots do not deposit pheromone in the cells that represent the walls of the environment, these cells receive an infinite value in the pheromone grid. Take into account the information provided by CA grids, the robots use this data to coordinate themselves throughout the environment. Furthermore, since the robots are using the IAS in the decision-making process, the pheromone information represents the probability of the robot move to a cell in a determined time step. The IAS is based on the Ant System, but in this case, the pheromone causes a repulsive behaviour on the robots, so they tend to spread in the environment, rather than staying close to each other. In other words, a high concentration of pheromone in a cell represents a low probability of this cell being chosen by a robot, on the other hand, a low concentration of pheromone, describes a high probability of this cell being selected.

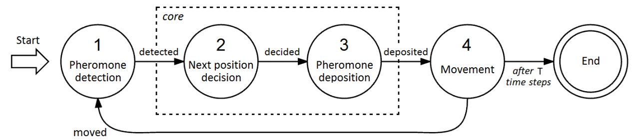
The IACA-DI model is divided into two levels: one is related to the behaviour of each robot and the other is related to the swarm's



(a) Physical grid.



(b) Pheromone grid.



(c) Robot behaviour through a FSM (Extracted from [35]).

Figure 1: Cellular Automata grids in a environment with 6 rooms and size (20×30) cells (Adapted from [36]).

global behaviour. The global behaviour of the swarm is related to robot-robot and robot-environment interactions, i.e., the emergence of a complex global behaviour originated through the local interactions between the robots and between the robots and the environment. In addition, the evaporation constant β is applied globally to ensure that the task is performed cyclically. In turn, a Finite State Machine (FSM) [34] was used to control the behaviour of each robot. Figure 1c illustrates the FSM with five states, which controls the robots at each time step of the simulation.

In the state Pheromone Detection, each robot reads the concentration of pheromone from its neighbourhood. Then, this reading is used in the Next Position Decision state to decide which one of the adjacent cells will be the destination of its next movement. In order to focus on more relevant parameters, the decision strategy Inertial Probabilistic is applied to all robots, since it has shown to be the most efficient strategy among the homogeneous compositions analysed in previous works [35].

The robots deposit pheromone in their current position and in their neighbourhood when the Pheromone Deposition state is activated. The pheromone deposition process is given by Equation 1 [34], which represents the amount of pheromone deposited in each cell x_{ij}, by each robot k, at each time step t. Finally, in the Movement state, the robots physically perform a movement in the environment, changing their position from the current cell to an adjacent cell in the neighbourhood.

$$\Delta_{ij}^{k} = (\psi_{max} - \psi_{ij}^{t}) \cdot \left[\alpha \cdot (\delta \cdot e)^{\eta \cdot \frac{r}{\pi}}\right] \tag{1}$$

The analysis of the IACA-DI model performed in our former work [34], was done through its application in the surveillance task. Surveillance is a frequent topic in research involving multi-robots and swarm robotics [8, 15, 17, 32]. In this task, robots should spread throughout the environment in the best possible way, seeking to monitor areas that were not visited for a long time or have never been visited. In this way, the environment must be monitored cyclically, ensuring that all areas are visited in the least practicable time interval.

2.2 Coloured Petri Nets

Modelling processes that interact with each other and consist of a large number of components with functionality similar to Petri nets, can result in multiple identical subnets [39]. This is due to the fact that, to differentiate two similar but distinct components, it is required to specify a subnet for each case, with identical structures. In addition, tokens in a Petri net usually represent objects or resources in a modelled process, which may have distinct attributes that cannot be represented by a single token. In order to solve this problem, [21] defined what is now known as the Coloured Petri Nets.

A CPN combines Petri nets theory with the modular functional programming language Standard ML [29] to obtain a graphical modelling language for concurrent processes. The formal basis for middling concurrency and synchronisation is provided by the Petri net, and the primitives for modelling data manipulation and creating compact and parameterizable models are provided by the functional programming language. Besides, the formation of a CPN is given by three parts (structure, inscriptions and statements): (i) the structure is a Petri net; (ii) inscriptions are variables associated with places, transitions and arcs; and (iii) statements are types of variables, functions, and operations on variables. The formal definition of a CPN [21] is given as follows.

Definition of Coloured Petri Net (CPN): A Coloured Petri Net associated with an initial tag is a septuple:

$$N_{CPN} = < R, C_{or}, C_{sc}, Var, G, E, M_0 > \quad (2)$$

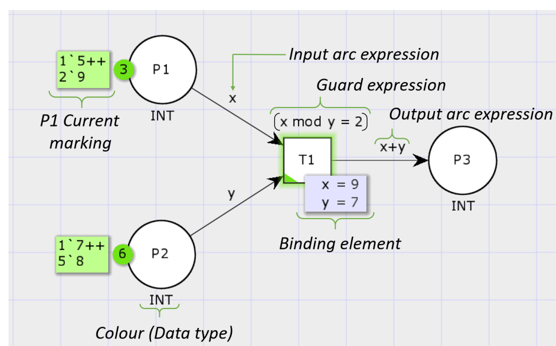
where:

- R is a Petri net;
- C_{or} is a nonempty set of types called colours;
- $C_{sc} : P \longrightarrow C_{or}$ is a colour function that associates each place with a subset of C_{or} (possible colours for a place);
- Var is a finite set of typed variables, such that $Type(v) \in C_{or}$ for all variables $v \in Var$;
- $G : T \longrightarrow exp$ is a guard function that associates each transition to a Boolean expression, such that $\forall t \in T | Type(G(t)) = Boolean \wedge Type(Var(G(t))) \subseteq C_{or}$;
- $E : (Pre \cup Post) \longrightarrow \{exp\}$ is a function of arc expressions, which associates to each arc 'a' C_{or} domain expression. The image of each arc expression must be a multiset with elements of the same colour associated with the place linked to the arc. Formally: $\forall a \in (Pre \cup Post) | Type(E(a)) = C_{or} \wedge Type(Var(E(a))) \subseteq C_{or}$;
- M_0 is the initial mark, which associates to each place $p \in P$ an expression whose result is a multiset over the colour set of places.

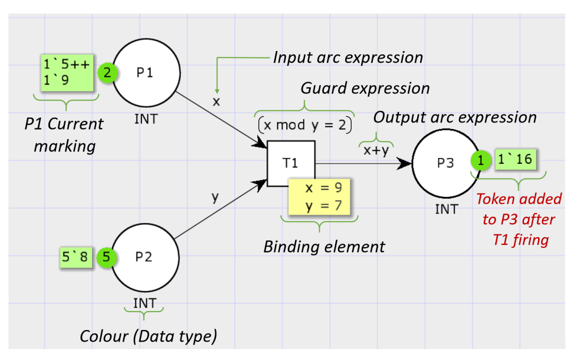
A place can only have tokens whose values respect the colour (domain) associated with this place. These tokens are represented by data structures that can contain information, allowing arcs to perform operations [21]. Arc expressions can contain constants, variables, functions and operations, defined in declarations to manipulate the information contained in the tokens. On the other hand, transitions determine the dynamics of CPN and may have "guard expressions" associated. These, in turn, are Boolean expressions that indicate the types of tokens that make it possible to activate a transition, thus building a type of firing constraint.

Guard and arc functions are responsible for imposing conditions for a token to move in the net, from one place to another. To understand the firing rules of a CPN, it is essential to be aware what is transition variables and binding elements. Transition variables are expressions associated with arcs that focus on transitions. A transition is enabled if it has enough tokens to satisfy the arcs inscriptions (if there is a binding element) and if its guarding is satisfied. This removes tokens from entry places and adds tokens to exit places, according to the expressions (inscriptions evaluation) of the arcs that connect a transition to their entry and exit places.

Figure 2 illustrates an example of CPN, detailing the states of the network before (Fig. 2a) and after (Fig. 2b) the firing of a transition. Figure 2a shows 3 places represented by $P1$, $P2$ and $P3$. These places contain type colours (integers) and the places $P1$ and $P2$ are associated with different types of tokens: $P1$ has one token of type 5 and two tokens of type 9, and $P2$ has one token of type 7 and five tokens of type 8. The three arcs present in the model contain expressions. In the arcs that come out of $P1$ and $P2$, the expressions



(a) CPN state before firing the transition.



(b) CPN state after firing the transition.

Figure 2: Example of CPN with a sensitised transition. Figure 2a shows the transition $T1$ sensitised, since both firing restrictions (tokens and guard) have been satisfied. Figure 2b shows the marking obtained after firing $T1$.

associated impose that only integer values can traverse these arcs (considering that the type of variables x and y was defined as Int). In addition, transition $T1$ has a guard function, i.e., $T1$ fires iff the division of the variable x by the variable y leaves 2 as the remainder. Finally, the arc that comes out of the transition $T1$ imposes that the token fired to $P3$ have to be the sum of the variables x and y, as is illustrated in Figure 2a. Firing the transition $T1$ (Fig. 2b), the place $P3$ receives a token with value 16, which in turn represents the sum of the tokens 9 and 7 from places $P1$ and $P2$, respectively. In this case, it can be seen that transition $T1$ can fire only once, as the guard function prevents any variable from being entered (the only variables that have satisfied this condition were variables 9 in $P1$ and 7 in $P2$, since $(9\ mod\ 7\ =\ 2)$.

Modelling, simulation and analysis of a process described in CPN strongly depend on the existence of computational tools that support the creation and manipulation of such models. Therefore, CPN Tools [22] was developed for editing, simulation and analysis of a CPN model. This tool allows users to work directly with the graphical representation of the net.

3 IACA-DI THROUGH CPNS

In order to carry out the formal analysis of the IACA-DI model (Sec. 2.1), it was modelled through CPNs (Sec. 2.2). This modelling was made in the CPN-Tools [22], which is a software for editing, simulating and providing state-space analysis of Petri nets. (available online under the GNU AGPLv3 license[1]).

Figure 3 represents the main CPN of the IACA-DI model. It is composed of 7 places (Environment, Cell mask, Robot, Ready, Detected, Decided and Deposited) and 6 transitions (Pre-processing, Pheromone detection, Cell decision, Pheromone deposition, Movement and Evaporation). Places 'Environment', 'Cell mask' and 'Robot' play the role of being tokens repositories. Thus, the system starts with all tokens inside these places, characterising its initial marking. In the surveillance task, there is no pre-established ending, thence this CPN does not have a specific a place for this purpose. However, to make the simulation feasible, it can be established a time-limit or a step-limit for each robot.

The environment in which the swarm is inserted is represented by a grid of cells (Sec. 2.1). This grid is described by a token inserted in the place 'Environment', which is accessed on-demand to decrease the intrinsic rigidity of synchronisation. By its turn, the place 'Robot' has tokens describing each robot of the swarm, along with their basics characteristics (e.g., id, current position and direction). Finally, tokens in the place 'Cell masks' represent the coordinates of each cell in the environment map. As the environment is a shared resource, it is necessary to regulate access concurrency. Thus, these mask tokens are used by the transition 'Pre-processing' in order to control which cells are in a busy state and which are not. Firing the transition 'Pre-processing' represents that a robot is starting o new cycle of movement, and the coordinates of the cells of its current position and its neighbourhood are now in a busy status. Whether the desired coordinates are being used by another robot, this transition is not sensitised.

[1]https://github.com/claudineyrt-research/IACADI-ColouredPetriNets-01

After a robot claims the cell coordinates of its current position and the cells of its neighbourhood, i.e. after the firing of the transition 'Pre-processing', a new token is added to the place 'Ready'. From this point on, the robot will start the necessary procedures to execute a movement, which in this case is to change from its current cell to a neighbouring one. These subsequent procedures have a direct relationship with the main cycle of the FSM described in Section 2.1. In order for the 'Pheromone detection' transition to be fired, it is necessary one token in the place of 'Ready' and one in place of 'Envir.', representing the availability of the environment map for reading. This reading consists of loading the information of the pheromone concentration in the neighbourhood according to the masks extracted from the place 'Cell mask'. Its firing generates one token in the place 'Detected', containing information of the respective robot and the pheromone concentration of the cells in its neighbourhood.

Transitions 'Cell decision', 'Pheromone deposition' and 'Movement' represent, respectively, the decision-making process of the target cell for the next movement, the pheromone deposits to signal the monitoring of a certain area, and the physical movement from the current cell to the target cell. In addition, these transitions represent the simplification of more complex CNPs, being called substitution transitions.

According to [22], substitution transitions are one of the morphisms used to obtain hierarchical networks. This type of transition makes it possible to divide the complexity of a net into subnets. As mentioned before, in this work, transitions 'Cell decision' (DCS - Fig. 4); 'Pheromone deposition' (DPS - Fig. 5); and, 'Movement' (MOV - Fig. 6) are substitution transitions, along with their respective subnets. Besides, the input and output interface between a substitution transition and its corresponding subnet is made through door-places. For instance, considering the subnet DCS_{st}, its input place is $Detected_p$ and its output place is $Decided_p$, i.e., $Detected_p \rightarrow DCS_{st} \rightarrow Decided_p$.

In addition to the modifications made by the robots, the evaporation makes changes in the concentration of pheromone. This is due to the fact that it represents a global natural process linked to the volatility of this substance in contact with the environment. Thus, the transition 'Evaporation' (Fig. 3) has a high firing priority (P_HIGH), i.e., whenever it is sensitised it will fire, regardless of the state of the other transitions. Moreover, in this work, its firing was defined by a discrete countdown. The evaporation rate is given by a constant β, and it is applied to all cells of the environment.

3.1 Subnet 'Cell decision' (DCS)

Figure 4 shows the expansion of the substitution transition DCS into a subnet. As mentioned, this transition is delimited by the door-places 'Detected' and 'Decided'. Besides, it corresponds to the decision-making process of the robots' next movement, considering the cells in their neighbourhood.

The decision process is divided into four main actions, represented in the CPN DCS (Fig. 4) by four transitions. The first one, 'release busy cells', removes, from the robots' neighbourhood, non-candidate cells to be the destination of the next movement, which are cells representing obstacles or cells occupied by other robots. Transition 'select choice strategy', defines which choice strategy will be applied (in previous works [35], different choice strategies were evaluated). In transition 'convert into probability', the pheromone concentration is transformed into percentages. In the last transition, 'spin roulette', a draw is carried out through a probabilistic roulette, similar to that applied in genetic algorithms.

3.2 Subnet 'Pheromone deposition' (DPS)

By its turn, Figure 5 illustrates the expansion of the substitution transition DPS, which models the stages of pheromone deposition. Communication between the robots is done indirectly by pheromone deposits throughout the environment. The pheromone concentration in a given cell indicates, proportionally, the time that cell was visited by a robot.

In the transition 'associate map and masks', the masks representing the current position and the neighbourhood of a robot are correlated to the cells of the environment. Once these cells have been defined, the pheromone deposition is carried out in the transition 'spread pheromone' (according to Eq. 1). Subsequently, the environment map is unloaded in the transition 'return updated map', and it becomes available for other robots. From this point, the robot will only need the masks of its current position and its next destination. Thus, in the transition 'return idle masks', the remaining masks are released and the deposition process is finalised.

3.3 Subnet 'Movement' (MOV)

The last substitution transition described in the main CPN describes the 'Movement' subnet. In this subnet, the movement of the robot is performed from a source cell to a target cell, which, in turn, was defined in the previous transitions.

In this case, the robot starts the movement process by acquiring the environment map to update its global position. This update is done in the transition 'update global position', and it considers the two masks (current and destination) brought from the previous steps. In a real environment, this step would represent the physical

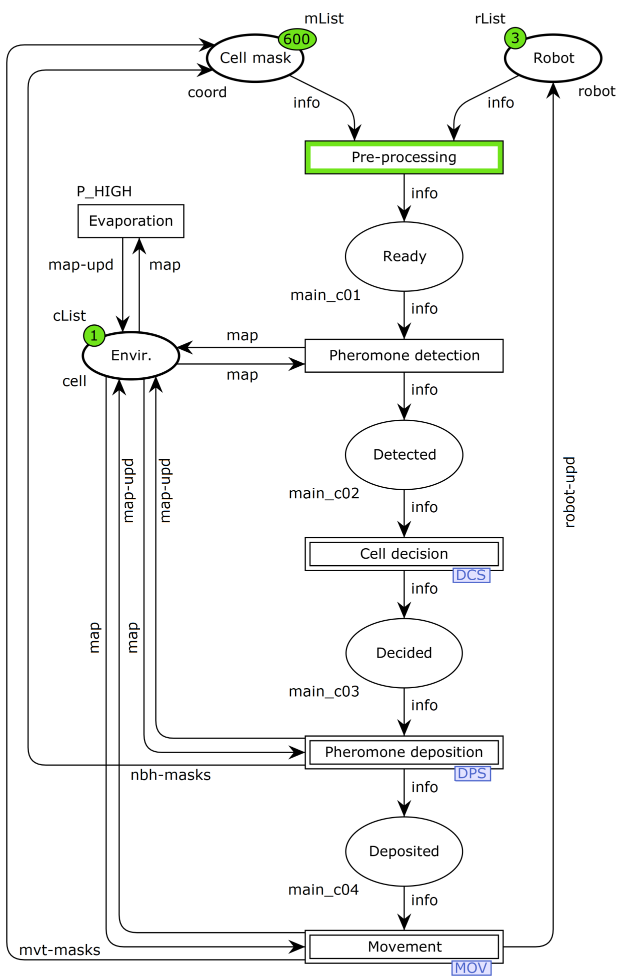


Figure 3: Main modelling network of the IACA-DI model [34] through Coloured Petri Nets in the CPN Tools platform.

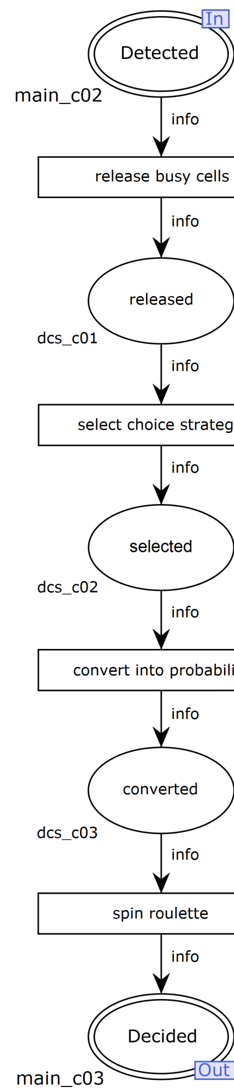


Figure 4: Expansion of the substitution transition "Cell decision" (DCS), modelled in the main CPN (Fig. 3).

movement of the robot. After updating its global position on the environment map, the robot updates its direction information and the room in which it is located. This update is carried out in the transition 'update robot structure'. In this second step, it is also necessary to load the map to define the current room. Once the local information has been updated, the environment map and the masks are released. The robot token, containing a new location, is returned to the place 'Robot', and it awaits a new firing of the transition 'Pre processing'.

4 MODEL VALIDATION ANALYSIS

The experiments described in this section aim to validate the IACA-DI model taking into account the formalisation providing by the CPNs. Thus, through the results, it is expected to verify the results obtained in previous works, mainly related to the effectiveness of the IACA-DI model. In addition, the main qualitative properties of Petri nets will be analysed (e.g., reachability, vivacity, limitability, conservativity and reversibility). It is noteworthy that, the IACA-DI model is intrinsically cyclical, i.e., the behaviour of robots repeats over time, finding that the model has the properties of reversibility vivacity is extremely important for its concise evolution.

The evaluation of the CPN was done through the State Space Analysis (SSA) [22]. This analysis is used to investigate control systems in a particular configuration. It is worth mentioning that, in the SSA, the number of parameters has a direct impact on the size of the state space. This growth is often exponential, causing an explosion in the size of the state space and characterising an important limitation of this technique. In other words, depending on the quantity and complexity of the model parameters, the complete analysis by SSA becomes impossible, due to the amount of memory and processing time required. In order to bypass this condition, the CPN Tools would generate a partial state space, i.e., a fragment of the real state space. However, this partial state space can not be used to verify formal properties, only to identify errors.

Some parameters of the IACA-DI model used in this work received the same values as those found in [34]: evaporation rate $\beta = 0.5\%$, vision radius $r_v = 1$, $\alpha = 0.5$, $\delta = 0.1$, $\eta = 2$, ($\mu = \nu = 30\%$). The environment of Figure 1a ((20×30) cells) was also selected. With the application of the inertial strategy, the cell that maintains the direction of the robot is twice as likely to be selected ($\gamma = 2.0$). On the other hand, in order to decrease the exponential explosion in the size of the state space, some of the parameters of the IACA-DI model were simplified. Different from [34], here the ψ variable was

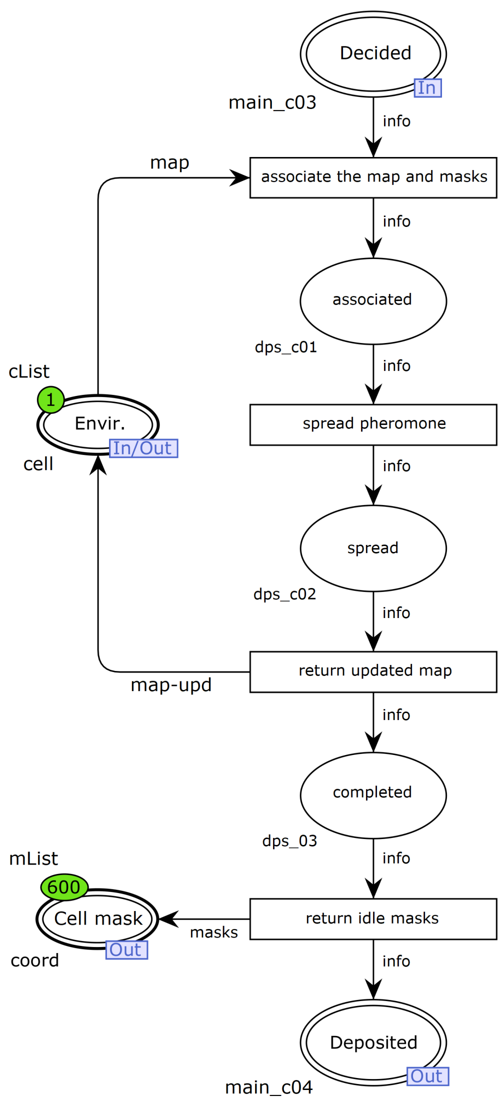


Figure 5: Expansion of the substitution transition "Pheromone Deposition" (DPS), modelled in the main CPN (Fig. 3).

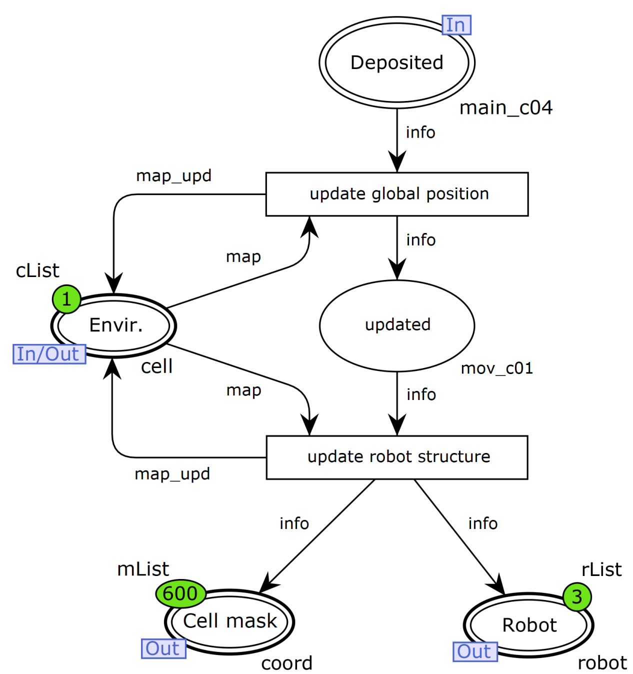


Figure 6: Expansion of the substitution transition "Movement" (MOV), modelled in the main CPN (Fig. 3).

```
Statistics
------------------------

  State Space
     Nodes:  12334
     Arcs:   12333
     Secs:   85
     Status: Full

  Scc Graph
     Nodes:  12334
     Arcs:   12333
     Secs:   0
```

Figure 7: SSA report: general statistics of the full state space for the CPN proposed through the IACA-DI model.

set to be $\{0 \leq \psi \leq 100 \parallel \psi \in \mathbb{N}\}$, therefore, $\psi_{max} = 100$. Besides, one robot ($N = 1$) and 1,000 time steps of simulation ($T = 1,000$) was used in this evaluation. Importantly, it is worth noting that simplified parameters do not affect the effectiveness of the IACA-DI model, since it inherits characteristics of swarm intelligence that foresees this type of drawback.

Considering that the CPN modelled does not have final places (see Section 2.2), due to the cyclicity inherited of the IACA-DI model, there was no need to apply the approach proposed by [38] in order to to test the net properties. Summarising, this approach consists of creating an extended Petri net, by adding an extra transition, connecting the end places to the start place of model, which would create a strongly connected Petri net.

Figures 7, 8 and 9, present the main parts of the SSA report (full report online[2]). The basic statistics of the model can be seen in Figure 7. It was obtained, applying SSA to the modelled CPN, a complete state space with 12,334 nodes and 12,333 arcs.

Figure 8 shows the results of the boundedness property. One can notice that the proposed CPN maintains stable a finite number of tokens. Therefore, it can be said that the CPN has the conservativity and limitability properties, since the amount of tokens never grows or goes down indefinitely. From a structural point of view, conservativity would be possible iff in all transitions the number of input arcs was equal to the number of output arcs. Nevertheless, because IACA-DI is a real system, general conservation of resources within a triggering range must be adopted. For instance, in the place “Pre-processing”, there are two input arcs and one output arc, which would result in a decrease in the number of total tokens. However, the resources were just combined to carry out a step of the task. At the end of this step, they separate and return to their original places. Conserving the number of tokens is extremely important to validate the model’s correctness when dealing with critical regions. The environment map, which contains information about the cells, is a data shared by all robots in the swarm, i.e., a critical region. To certify conservativity, is to prove that the robots are correctly updating the information in the map of the environment.

Finally, part of the state space report specifying home, liveness and fairness properties, is shown in Figure 9 . According to [22], one can define these properties as follows: (i) a home marking as

[2] https://github.com/claudineyrt-research/IACADI-ColouredPetriNets-01

```
Boundedness Properties
----------------------------------------------

 Best Integer Bounds
                             Upper      Lower
    DCS'freeNbh 1            1          0
    DCS'phNbh 1              1          0
    DCS'tyNbh 1              1          0
    DPS'dpsPH 1              1          0
    DPS'iNBH 1               1          0
    DPS'setPH 1              1          0
    MOV'MapUp 1              1          0
    main'CoordMap 1          600        591
    main'Decided 1           1          0
    main'Deposited 1         1          0
    main'Detected 1          1          0
    main'Envir 1             1          0
    main'Ready 1             1          0
    main'Robot 1             1          0
    main'TimeSteps 1         1          1
```

Figure 8: SSA report: boundedness properties of the full state space for the CPN proposed through the IACA-DI model.

a marking which can be reached from any reachable marking, i.e., regardless the firing sequence, the home marking will always be reached; (ii) a Petri net is said to be alive iff, for any marking, there is at least one firing-enabled transition; and, (iii) given any two transitions, it can be said that they are in bounded-fair relation, iff the maximum number of times that either one can fire while another is not firing is bounded.

Results of Figure 9 show one home marking and one dead marking in the liveness properties, both with number 12334. This was an expected outcome, since, to create a complete state space, there is one robot running the task and the simulation was limited to 1,000 time steps ($T = 1,000$). As can be seen in the state space statistics (Fig. 7), 12334 was the total number of nodes created. Therefore, it is possible to conclude that home marking and dead marking only happened due to time limitations and number of robots (it would also be possible to concluded that the network is reversible, since at each cycle it returns exactly to its initial state). In a real scenario, where these limitations are not applied, it is expected that such markings will not exist. Which is an expected feature, especially when the task assigned to the swarm of robots is the surveillance of environments, since this it is performed without a stopping point. Finally, it is worth noting the non-occurrence of infinite sequences in the fairness property, one of the main results of this analysis. This result shows that the sequence of steps established in the IACA-DI model are happening according to the established FSM (see Fig. 1c), i.e., there is no possibility of certain transitions to fire indefinitely, without first go through each stage of the model.

5 CONCLUSIONS AND FUTURE WORK

This work has proposed the application of Coloured Petri Nets in the formalisation, modelling and analysis of the coordination model for swarm robotics IACA-DI [34]. The main objective of the IACA-DI model is to coordinate a swarm of robots in covering, foraging and, especially, in surveillance tasks. Among the main characteristics of this work, it is worth highlighting: (i) despite discretization

```
Home Properties
------------------------------------------

 Home Markings
    [12334]


Liveness Properties
------------------------------------------

 Dead Markings
    [12334]

 Dead Transition Instances
    None

 Live Transition Instances
    None


Fairness Properties
------------------------------------------
    No infinite occurrence sequences.
```

Figure 9: SSA report: home, liveness and fairness properties of the full state space for the CPN proposed through the IACA-DI model.

and formalisation, modelling through Petri nets showed a positive adaptation to swarm robotics, maintaining the fundamental characteristics of the latter [5]; (ii) the Coloured Petri net resulting from the IACA-DI modelling, can be scaled for swarms with different amounts of robots and for different input environments; (iii) being the environment a critical region, resource blocking, an intrinsic characteristic of Petri nets, allowed the pheromone readings and deposits in the environment to be synchronised automatically; (iv) synchronism is not strong regarding the net, since different robots can be in different places of the net at the same time; and (v) the Coloured Petri nets allowed the analysis of all possible states of the system, which would be a difficult objective to achieve in common robotic simulators since the number of states grows exponentially.

According to the obtained results, it was possible to verify that the analysis and results presented in previous works, in the majority of the cases through empirical observations, that the model proved to fulfil its expected characteristics. Thus, we can conclude that: (i) the IACA-DI model is adequate to coordinate a swarm of robots in the proposed tasks; (ii) the model behaves cyclically, finding no deadlock points during the execution of the task, which is an extremely important feature for surveillance and foraging tasks; e (iii) without any direct communication between the robots, using just an indirect communication through the deposition of pheromone in the environment, it was possible the execute the task with promising outcomes.

Modelling the IACA-DI through Petri nets made it possible to confirm the robustness and dynamism of this technique. This is mainly due to the formalisation brought by the Petri nets and the possibility of evaluating a large number of different states. Therefore, as future work, we intend (i) to continue the analysis of the IACA-DI model based on the initial results presented in this work, focusing mainly on the variation of input parameters (e.g., different environments and swarms with a larger number of robots); (ii) work to improve the computational efficiency of Petri net modelling, which appears to be promising, especially with regard to the environment map that is shared by all robots of the swarm; (iii) search for a way to generalise Petri nets modelling for different types of multi-robots and robot swarms coordinating systems, allowing the comparison and systemic evaluation between the systems; and (iv) investigate other applications of Petri nets in different discrete bio-inspired systems, such as cryptography, task scheduling and evolutionary system problems.

AVAILABILITY OF ARTEFACTS

The authors support Open Science practices, which seek to promote transparency, replicability and reproducibility of research works. Therefore, the CPN model proposed here is available online under the GNU AGPLv3 license at the following address: https://github.com/claudineyrt-research/IACADI-ColoredPetriNets-01

ACKNOWLEDGMENTS

The authors thank the following Brazilian agencies for their financial support and scholarships: FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and CAPES (Coordenação de Aperfeiçamento de Pessoal de Nível Superior).

REFERENCES

[1] Franciny M. Barreto and Stéphane Julia. 2017. Modeling of video games using workflow nets and state graphs. In *Brazilian Symposium on Software Engineering (SBES)*. Association for Computing Machinery, Fortaleza, Brazil, 261–266.
[2] Kamalika Bhattacharjee, Nazma Naskar, Souvik Roy, and Sukanta Das. 2020. A survey of cellular automata: types, dynamics, non-uniformity and applications. *Natural Computing* 19, 2 (2020), 433–461.
[3] Tilemachos Bontzorlos and Georgios Ch Sirakoulis. 2017. Bioinspired algorithm for area surveillance using autonomous robots. *International Journal of Parallel, Emergent and Distributed Systems* 32, 4 (2017), 368–385.
[4] Evangelos Boukas, Ioannis Kostavelis, Antonios Gasteratos, and Georgios Ch Sirakoulis. 2015. Robot guided crowd evacuation. *IEEE Transactions on Automation Science and Engineering* 12, 2 (2015), 739–751.
[5] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. 2013. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence* 7, 1 (2013), 1–41.
[6] Maria Paola Cabasino, Alessandro Giua, and Carla Seatzu. 2013. Diagnosability of discrete-event systems using labeled Petri nets. *IEEE Transactions on Automation Science and Engineering* 11, 1 (2013), 144–153.
[7] M Caccia, P Coletta, G Bruzzone, and G Veruggio. 2005. Execution control of robotic tasks: a Petri net-based approach. *Control Eng. Pract.* 13, 8 (2005), 959–971.
[8] Rodrigo Calvo, Janderdeson Rodrigo de Oliveira, Mauricio Figueiredo, and Roseli Ap. Francelin Romero. 2011. Bio-inspired coordination of multiple robots systems and stigmergy mechanims to cooperative exploration and surveillance tasks. In *Int. Conf. on Cybernetics and Intelligent Syst. (CIS)*. IEEE, Qingdao, China, 223–228.
[9] Alexandre Campo and Marco Dorigo. 2007. Efficient multi-foraging in swarm robotics. In *Eur. Conf. on Artif. Life (ECAL)*. Springer, Lisbon, Portugal, 696–705.
[10] Nabendu Chaki and Bidyut Biman Sarkar. 2010. Virtual data warehouse modeling using petri nets for distributed decision making. *Journal of Convergence Information Technology* 5, 5 (2010), 8–21.
[11] Hugo Costelha and Pedro Lima. 2007. Modelling, analysis and execution of robotic tasks using petri nets. In *International Conference on Intelligent Robots and Systems*. IEEE, San Diego, CA, USA, 1449–1454.
[12] Francisco A. X. da Mota, Matheus Xavier Rocha, Joel J. P. C. Rodrigues, Victor Hugo C. de Albuquerque, and Auzuir Ripardo de Alexandria. 2018. Localization and navigation for autonomous mobile robots using Petri nets in indoor environments. *IEEE Access* 6 (2018), 31665–31676.
[13] Gina M. B. de Oliveira, Reslley G. O. Silva, Laurence R. do Amaral, and Luiz G. A. Martins. 2018. An Evolutionary-Cooperative Model Based on Cellular Automata

and Genetic Algorithms for the Navigation of Robots Under Formation Control. In *Braz. Conf. on Intelligent Syst. (BRACIS)*. IEEE, Sao Paulo, Brazil, 426–431.

[14] Leiliane P. de Rezende, Stéphane Julia, and Janette Cardoso. 2016. Possibilistic WorkFlow nets for dealing with cancellation regions in business processes. In *Int. Conf. on Enterprise Information Systems (ICEIS)*. SciTePress, Roma, Italy, 126–133.

[15] Evandro Luis S. Falleiros, Rodrigo Calvo, and Renato P. Ishii. 2015. PheroSLAM: A Collaborative and Bioinspired Multi-agent System Based on Monocular Vision. In *Computational Science and its Applications (ICCSA)*. Springer, Cham, 71–85.

[16] Joslaine Cristina Jeske de Freitas, Stéphane Julia, and Leiliane Pereira de Rezende. 2015. Resource Allocation Mechanisms and Time Constraint Propagation Techniques in Fuzzy Workflow Nets. In *International Conference on Enterprise Information Systems*. Springer International Publishing, Cham, 214–235.

[17] Dean F Hougen, Saifallah Benjaafar, Jordan C Bonney, John R Budenske, Mark Dvorak, Maria Gini, Howard French, Donald G Krantz, Perry Y Li, Fred Malver, et al. 2000. A miniature robotic system for reconnaissance and surveillance. In *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, San Francisco, USA, 501–507.

[18] Hesuan HU and MengChu ZHOU. 2014. A Petri net-based discrete-event control of automated manufacturing systems with assembly operations. *IEEE Transac. on Control Syst. Technology* 23, 2 (2014), 513–524.

[19] Konstantinos Ioannidis, G Ch Sirakoulis, and Ioannis Andreadis. 2011. Cellular ants: A method to create collision free trajectories for a cooperative robot team. *Robotics and Autonomous Syst.* 59, 2 (2011), 113–127.

[20] Marcin Jamro, Dariusz Rzonca, and Wojciech Rząsa. 2015. Testing communication tasks in distributed control systems with SysML and Timed Colored Petri Nets model. *Computers in Industry* 71 (2015), 77–87.

[21] Kurt Jensen. 1981. Coloured Petri nets and the invariant-method. *Theoretical computer science* 14, 3 (1981), 317–336.

[22] Kurt Jensen, Lars Michael Kristensen, and Lisa Wells. 2007. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer* 9, 3 (2007), 213–254.

[23] Stéphane Julia, Fernanda Francielle de Oliveira, and Robert Valette. 2008. Real time scheduling of Workflow Management Systems based on a p-time Petri net model with hybrid resources. *Simulation Modelling Practice and Theory* 16, 4 (2008), 462–482.

[24] Marius Kloetzer and Cristian Mahulea. 2014. A Petri net based approach for multi-robot path planning. *Discrete Event Dynamic Systems* 24, 4 (2014), 417–445.

[25] Xiaoou Li, Joselito M. Medina, and Sergio V. Chapa. 2007. Applying Petri nets in active database systems. *IEEE Trans. Syst., Man, and Cybern.* 37, 4 (2007), 482–493.

[26] Luke Ludwig and Maria Gini. 2006. Robotic swarm dispersion using wireless intensity signals. In *Distributed Auton. Robotic Syst. 7*. Springer, Tokyo, 135–144.

[27] Luiz G. A. Martins, Rafael da P. Cândido, Mauricio C. Escarpinati, Patricia A. Vargas, Gina M. B. Oliveira, et al. 2018. An improved robot path planning model using cellular automata. In *Annual Conference Towards Autonomous Robotic Systems (TAROS)*. Springer, Cham, 183–194.

[28] Flávio Félix Medeiros and Stéphane Julia. 2017. Constraint Analysis based on Energetic Reasoning Applied to the Problem of Real Time Scheduling of Workflow Management Systems. In *19th International Conference on Enterprise Information Systems (ICEIS)*. SciTePress, Porto, Portugal, 373–380.

[29] Robin Milner, Mads Tofte, Robert Harper, and David MacQueen. 1997. *The definition of standard ML: revised*. MIT press, Cambridge, Massachusetts.

[30] Dayal Ramakrushna Parhi and Jagadish Chandra Mohanta. 2011. Navigational control of several mobile robotic agents using Petri-potential-fuzzy hybrid controller. *Applied Soft Computing* 11, 4 (2011), 3546–3557.

[31] Ameer L Saleh, Mohammed J Mohammed, Ahmed Sabri Kadhim, Hana'a M Raadthy, and Hesham J Mohammed. 2018. Design fuzzy neural petri net controller for trajectory tracking control of mobile robot. *International Journal of Engineering & Technology* 7, 4 (2018), 2256–2262.

[32] Martin Saska, Jan Vakula, and Libor Přeućil. 2014. Swarms of micro aerial vehicles stabilized under a visual relative localization. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Hong Kong, China, 3570–3575.

[33] S. P. Sharma, N. Sukavanam, Naveen Kumar, and Ajay Kumar. 2010. Reliability analysis of complex robotic system using Petri nets and fuzzy lambda-tau methodology. *Engineering Computations* 27, 3 (2010), 354–364.

[34] Claudiney R. Tinoco, Danielli A. Lima, and Gina M. B. Oliveira. 2017. An improved model for swarm robotics in surveillance based on cellular automata and repulsive pheromone with discrete diffusion. *Int. Journal of Parallel, Emergent and Distributed Systems (IJPEDS)* 34 (2017), 53–77.

[35] Claudiney R. Tinoco and Gina M. B. Oliveira. 2018. Pheromone Interactions in a Cellular Automata-Based Model for Surveillance Robots. In *Int. Conf. on Cellular Automata (ACRI)*. Springer, Cham, Como, Italy, 154–165.

[36] Claudiney R. Tinoco and Gina M. B. Oliveira. 2019. Heterogeneous Teams of Robots using a Cellular Automata-Based Model for Surveillance Task. In *Cong. on Evolutionary Computation (CEC)*. IEEE, Wellington, NZ, 154–165.

[37] Lan Anh Trinh, Mikael Ekström, and Baran Cürüklü. 2019. Petri net based navigation planning with dipole field and dynamic window approach for collision avoidance. In *6th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, Paris, France, 1013–1018.

[38] Wil Van Der Aalst, Kees Max Van Hee, and Kees van Hee. 2004. *Workflow management: models, methods, and systems*. MIT press.

[39] Willibrordus Martinus Pancratius Van der Aalst. 1992. Timed coloured Petri nets and their application to logistics. *Master Thesis* (1992), 267.

[40] Zhongyang Zheng and Ying Tan. 2013. Group explosion strategy for searching multiple targets using swarm robotic. In *Congress on Evolutionary Computation (CEC)*. IEEE, Cancun, Mexico, 821–828.