

# Swarm Approaches for the Patrolling Problem, Information Propagation vs. Pheromone Evaporation

Hoang-Nam Chu<sup>1,2</sup>, Arnaud Glad<sup>1</sup>, Olivier Simonin<sup>1</sup>, François Sempé<sup>2</sup>,  
Alexis Drogoul<sup>3,2</sup>, François Charpillet<sup>1</sup>

<sup>1</sup>MAIA, INRIA Lorraine, Campus scientifique, BP 239, 54506 Vandœuvre-lès-Nancy, France

<sup>2</sup>Institut Francophone pour l'Informatique, Hanoi, Vietnam

<sup>3</sup>IRD - Institut de Recherche pour le Développement, Bondy, France

{hoangnam.chu, arnaud.glad, olivier.simonin, francois.charpillet}@loria.fr  
francois@ifi.edu.vn, drogoul@mac.com

## Abstract

*This paper deals with the multi-agent patrolling problem in unknown environment using two collective approaches exploiting environmental dynamics. After specifying criteria of performances, we define a first algorithm based only on the evaporation of a pheromone dropped by reactive agents (EVAP). Then we present the model CLInG [10] proposed in 2003 which introduces the diffusion of the idleness of areas to visit. We systematically compare by simulations the performances of these two models on growing-complexity environments. The analysis is supplemented by a comparison with the theoretical optimum performances, allowing to identify topologies for which methods are the most adapted.*

**Keywords:** Multi-agent patrolling, reactive multi-agents system, digital pheromones.

## 1. Introduction

Patrolling consists in deploying a set of agents (robots) in an environment in order to visit regularly all the accessible places [5].

This problem was studied in recent years according to centralized, heuristic and distributed approaches, but always within a discrete representation of the environment, i.e. a graph. A vertex is a predetermined place that should be visited and an edge is a valid path linking two places. Thus, various work based on graph search algorithms have been proposed, often deriving from the problem of the traveling salesman (cf. [1] for a presentation of these various techniques and their comparison). For instance Lauri and Charpillet [4] proposed a solution relying on ACO

algorithms (ants colonies optimization) which requires a representation of the environment through a graph. There are also approaches based on learning techniques (e.g. [8]). They consist in computing offline an optimal multi-agent path, which is then carried out online in the considered environment. Consequently, this type of solution is not able to self-adapt to online changes of the problem/environment, such as variations of the number of agents or moves of obstacles, etc.

Moreover, these approaches are subject to *combinatory explosion* when the graph size becomes important (several hundreds of nodes) or when the number of deployed agents increases.

However, nowadays, many concrete applications present the patrolling problem on large spaces, known or *unknown*, with a significant number of agents (drones deployed to supervise a strategic place, patrolling of buildings by mobile robots, etc.).

So, to deal with such a configuration of the problem (unknown environments) we think that swarm intelligence could be an efficient approach. It is generally based on the marking of the environment, inspired by the ants' pheromone drop, which defines an indirect calculation and means of communication among the agents [2].

These digital pheromones rely on two processes calculated by the environment: the diffusion and the evaporation of information (pheromone's quantity).

The diffusion process enables the propagation of the information by effect of vicinity, while evaporation allows removing gradually the information.

Sempé et al. [10] proposed in 2003 an algorithm named CLInG exploiting the propagation of information, which is close to the diffusion process, showing the interest of an approach based on an active environment. However this approach appears relatively expensive as it exploits processes of propagation and

of counting in the environment (idleness evaluation). Thus we propose in this article another algorithm based only on the evaporation of a digital pheromone laid down by agents, named EVAP model.

Our objective is to compare these two collective techniques, which are based on an active environment, but having different complexity, in order to better understand the functioning and the performances of these stigmergic principles.

The article is organized as follows. Section 2 defines the patrolling problem and gives the performance criteria. In section 3 we present the EVAP model, and in section 4 the CLInG model. Section 5 presents experimentation of both models, where they are compared and analyzed on environments with an increasing complexity. In section 6 we synthesize these results to draw the interest of the two approaches. Finally this work ends with a conclusion and presents some perspectives.

## 2. The multi-agent patrolling problem

Patrolling consists in deploying several agents, generally in a fixed number, in order them to visit the strategic places of an area periodically. It aims at obtaining reliable information, seeking objects, watching over places in order to defend them for any intrusion, etc.

An efficient patrol in an environment (possibly dynamic) needs the delay between two visits of the same place to be minimal. Related works on multi-agent patrol strategies consider that the environment is *known*, two-dimensional and that *it can be reduced to a graph*  $G(V, E)$  ( $V$  the set of the nodes to be visited,  $E$  the arcs defining the valid paths between nodes).

Several criteria may be used in order to evaluate the efficiency of a patrolling strategy. We adopt those based on the node idleness computing, which corresponds to the time elapsed from the last visit by an agent. The following criteria, relative to idleness and presented in [5], can be calculated either at the node level or at the graph level:

- Instantaneous Node Idleness (INI): Number of time steps elapsed while the node is not visited. This criterion is computed for every node.
- Instantaneous Graph Idleness (IGI): Average of Instantaneous Node Idleness of each node at a given time. This criterion is computed at the graph level
- Instantaneous Worst Idleness (IWI): Highest INI computed at a given time, graph level criterion.

As it was emphasized in the introduction, in this paper we tackle the multi-agent patrolling problem in unknown environments, i.e. for which we do not have a graph of the areas to visit. So, we consider the space to patrol as a vast empty grid, which have a granularity

defined by the area an agent can perceive at a given time.

More precisely, the environment is a matrix where each cell may be free, occupied by an agent or unreachable (obstacle/wall). So two agents cannot occupy the same cell at the same moment.

Patrolling is simulated with a constant time step that enables every agent to do an elementary action: move to one of the four neighboring cells. The definition of idleness presented above for the nodes of graphs is retained and applied to each cell that must be visited.

This paper focuses on the generic patrolling problem which consists in minimizing the average idleness of accessible cells. We also take into account to minimize the worst idleness.

## 3. EVAP: model based on deposit and evaporation of information

### 3.1. Evaporation process

The swarm intelligence principle, inspired by the study of social insects [2], is based on the collective organization of the agents through their indirect communications in the environment. These communications are based on the deposit and the diffusion of a chemical substance which allows agents to cooperate through the environment. So, the computational model, called *digital pheromones*, needs to define an active environment allowing both evaporation and diffusion process. This indirect way of communication is particularly interesting to deal with tasks in initially unknown environments (e.g. foraging [6], path-planning [7], coverage, exploration ...).

So, we propose a new algorithm for multi-agent patrolling in unknown environment, named EVAP, which relies on the evaporation process of a pheromone dropped by the agents. This model only uses the evaporation process to exploit the remaining quantity of pheromone as an indicator of the time elapsed since the last visit of a cell (representing the idleness). Thus, we define the agents behavior as a descent of the pheromone gradient (i.e. moving towards the cells containing the less pheromone quantity).

The evaporation process of the pheromone in a cell is expressed by the following geometrical series on the quantity:

$$q_{n+1} = q_n * (1 - \rho)$$

This process enables the creation of an oriented gradient following the cells visit chronology. It only requires  $\rho \in (0, 1)$  and  $q_0 > 0$ . In fact  $q_n$  decreases for any  $\rho$  value in  $(0, 1)$ , so the model is independent from  $\rho$  parameter. In the experiments  $\rho = 0.001$ .

### 3.2. Algorithm

The following-gradient-flow behaviour allows agents to explore the most formerly visited (or not visited at all) cells. An agent can perceive over its four neighbouring cells (noted Neighbourhood in the algorithms) for which it can read the pheromone quantity. Then, it moves towards the one containing the minimum value and drops  $Q_{\max}$  value.

In case several neighbouring cells contain the same amount of pheromone, the agent chooses to move randomly among them. However, to avoid too erratic trajectories, if the current direction can be kept, the agent chooses, according to a probability  $p$ , to maintain it.

#### EVAP Agent ALGORITHM

```

m = min(PheroQ (Neighborhood))
For each cell c of Neighborhood Do
  If PheroQ(c) = m Then
    NeighborsList ← NeighborsList + c
  EndIf
EndFor
nextCell ← cell to which we are heading
If nextCell ∈ NeighborsList
  and random(1) < p Then
    moveTo(nextCell)
  Else
    moveTo(random(NeighborsList))
  EndIf
DropPheromones(Qmax)

```

#### EVAP Environment ALGORITHM

```

For each cell c of environment Do
  If PheroQ(c) > 0 Then
    ComputeEvapPhero(c)
  EndIf
EndFor

```

The EVAP model can be seen as an extension of the Yanovski et al. algorithm [12] to grid environments. Indeed, the algorithm in [12] uses a pheromone which is dropped on the *edges* of a given graph necessarily both *oriented* and *eulerian*. The principle of the EVAP agent behavior is the same as the gradient descent described in [12].

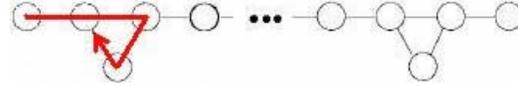
### 3.3. Interest of the swarm intelligence

In an early work on the graph coverage problem, Wagner et al. [11] emphasized drawbacks of reactive approaches using marking of visited nodes. They showed that sub-optimal exploration can be generated for certain topologies when using only one agent. This problem comes from the very limited perceptions of agents. It can be explained as following.

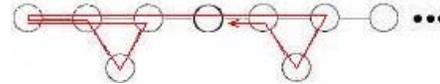
Suppose that an agent has visited three consecutive nodes of the following graph while dropping pheromones. It now faces to the choice between two not yet explored nodes:



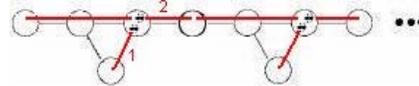
It is obvious that a random behavior may conduct to the following “wrong” choice:



The initial marking created a gradient that forces the agent to cross already visited cells before going back and continuing its exploration task. This phenomenon may happen every time a similar choice happens:



A means to avoid this kind of problem is to give to agents a perception further than the immediate nodes, in order to avoid paths returning to already visited cells. For this purpose, we will see that CLInG uses the environment to attract agents towards the not-yet-visited nodes. For EVAP, we keep a limited perception as we assume that cooperation between agents can reduce the problem. If two agents visit in a short delay a node which requires a choice between two possible routes; if the first one makes the “bad” choice, the second one will necessary make the good choice, i.e. to continue the exploration because it will follow a not yet explored path:



We notice that the more agents there are, the most this drawback will be attenuated (see section 5). We show through the experiments conducted in section 5 that this limit exists only for specific topographies. In those cases, CLInG proves to be a good solution.

## 4. CLInG: model based on information diffusion

### 4.1. Approach based on idleness diffusion

Sempé et al. [9] [10] proposed a patrolling algorithm which supposes that agents are reactive (like EVAP) and that the environment calculates two following information:

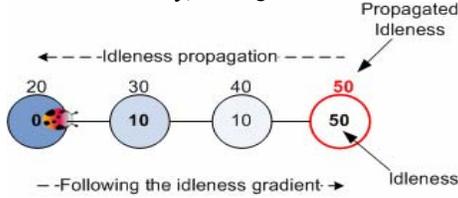
- the Idleness of each cell
- the diffusion of maximal Idleness values

At each iteration, the environment calculates the idleness of each accessible cell by incrementing its value by 1 unity. The idleness of a cell is reset to zero when it is visited.

The originality of algorithm CLInG is adding second information into the environment by the diffusion of the maximum idleness. Propagating makes

a second gradient that guides the agents to cells of interests (the most formerly visited).

More formally, a cell  $i$  carries a propagated idleness  $OP_i$  besides its individual idleness  $O_i$ . The gradient created by the propagated idleness is shared by the whole collectivity, cf. Figure 1.



**Figure 1. Propagated Idleness**

The propagated idleness of a cell depends on the propagated idleness of its neighbors and its individual idleness. It is equivalent to a function that takes into account the idleness and the presence of agents on the way.

$$OP_i = \max[O_i, \max(f(i, j))]$$

with  $j$  the neighboring cells of  $i$ , and  $f$  the propagation function :

$$f(i, j) = OP_j - \alpha - \beta \cdot I(j)$$

$$\text{If } OP_j - \alpha - \beta \cdot I(j) \geq OP_{\min}$$

$$= OP_{\min} \quad \text{If } OP_j > OP_{\min}$$

$$= OP_j - 1 \quad \text{Otherwise}$$

$\alpha$  is the coefficient of propagation. Its value is important (for example 30 in the experiments) in order to create a short distance gradient that does not risk to attract all the agents to a maximum idleness cell.

$I(j)$  is the interception function that stops a propagation whenever it meets an agent.  $I(j)$  equals to 1 if there is an agent in cell  $j$ , otherwise 0. This factor also restrains the gathering of agents from the same way (the order of magnitude of  $\beta$  is 10), cf. details in [9] [10].

$OP_{\min}$  is a threshold that assures the propagated idleness forms always a gradient and remains positive due to the fact that the individual idleness is always positive. The behavior of each agent consists in following the gradient of maximal idleness (cf. illustration Figure 1). This is a dual approach of the previous algorithm (EVAP), but this time, the information in the neighboring cells can come from further cells.

The propagation of the maximum idleness allows exploiting the inherent environment's properties and to transform objective information into subjective one which can be used directly by the agents. The algorithm provides thus an organization among the agents according to the distribution of the idleness in the environment.

## 4.2. Algorithm

### CLInG Agent ALGORITHM

```

m = max(Propagated_I (NeighboringCells))
For each cell c of NeighboringCells Do
  If Propagated_I (c) = m Then
    ListNeighbors ← ListNeighbors + c
  End If
End For
moveTo (Random(ListNeighbors))
Idleness(currentCell) ← 0

```

### CLInG Environment ALGORITHM

```

For each cell c of Environment Do
  Calculate Idleness(c)
End For
For each cell c of Environment Do
  Calculate Propagated_I(c)
End For

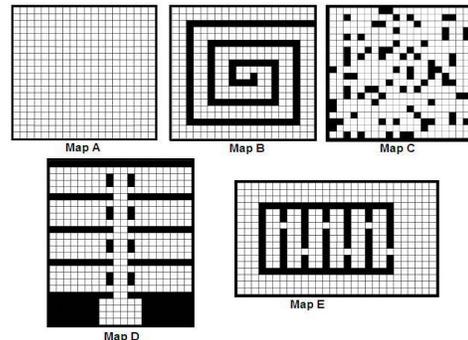
```

## 5. Simulations and analysis

### 5.1. Methodology

The simulations were implemented in NetLogo. The conditions of simulation, perceptions and moves are strictly the same for the two algorithms.

The experiments of the 2 models were performed in 5 environments with growing complexity, taken or adapted from [1] and [11], cf. Figure 2 (obstacles in black). Topology A is an open-field environment which means that agents are free for any movement. Environment B is a spiral enabling a corridor with dead ends. Environment C allows constraining the environment by a density of obstacle cells, generated randomly (we used a density of 20%, without isolated cells). Environment D represents a corridor overlooking 8 rooms. At last, E presents 6 rooms which have imbricate entries, that we name 6-rooms. More generally we define the n-rooms problem as n rooms with imbricate entries.



**Figure 2. Experimental topologies**

We tested the algorithms with different population sizes by doubling systematically the number of agents: 1, 2, 4, 8, 16, 32 and 64. We aim at evaluating performances and collective skills of both models.

Every simulation is executed during 3000 iterations (4000 for environments D and E) and 10 times to compute means values.

## 5.2. Simple environments

### Non-obstacle environment (Map A)

Figure 3 shows performances with 8 agents, during the first 1000 iterations, in a 20x20 cells environment (agents are initially randomly located). This plot illustrates the average idleness and the worst idleness of both studied methods (CLInG and EVAP). We note apparently that both average idleness values are rapidly stable and identical, and moreover, very close to the optimum theoretical value (represented by the horizontal line).

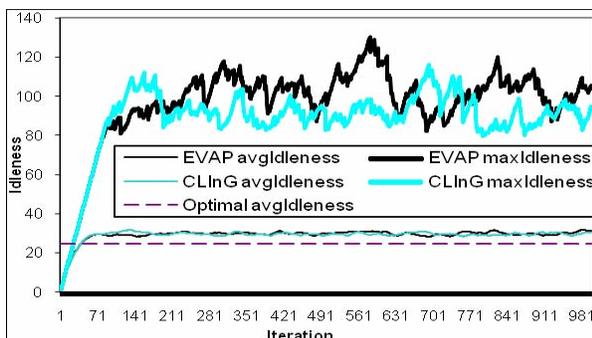


Figure 3. Non-obstacle topology, 8 agents, 1000 it.

The theoretical optimum idleness values are calculated as follows: Let  $c$  be the number of accessible cells of the environment. Considering one agent moving towards a new cell at each iteration, then it visits all cells in  $c-1$  iterations. Thus, the idleness of the departure cell will reach  $c-1$ . For  $n$  agents, the optimum maximum idleness is  $(c/n)-1$ . The optimum average idleness is therefore  $((c/n)-1)/2$  (as idleness values are linearly distributed between 1 and the optimum value).

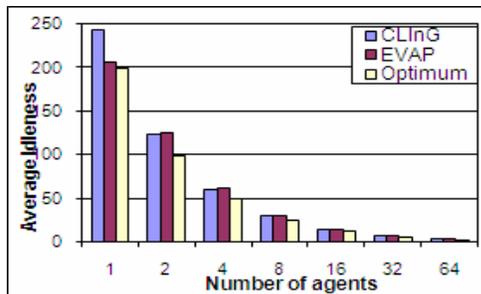


Figure 4. Non-obstacle topology, average IGI

Figure 4 illustrates the average idleness of both methods for a variation of the agents number. In both cases, the doubling of agents improves the performances linearly. For each configuration, we notice that the obtained values almost correspond to

the theoretical optimum value. It is clear that CLInG and EVAP have the same effectiveness for this topology, except the single agent case. This result is also verified with bigger environments.

### Spiral environment (Map B)

Figure 5 illustrates the average and the worst idleness values with 4 agents in a 20x20 cells spiral environment, over 3000 iterations (agents are initially randomly located). Regarding to the worst idleness values, EVAP is slightly better despite the fact that it does not become stable. As in the previous topology, the average idleness values are very close to the optimum value and relatively stable.

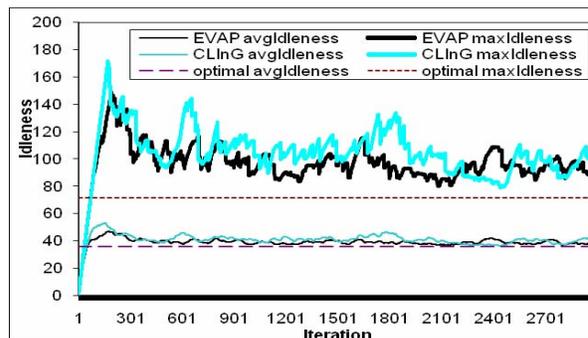


Figure 5. Spiral topology, 4 agents 3000 iteration

Figure 6 shows that EVAP is a bit more efficient than CLInG on average idleness for a small number of agents (up to 4). This is clear for one and two agents, where EVAP almost attains the theoretical optimum value. It is important to note that for algorithm EVAP, we obtained this optimum performance for  $p=1$ , the probability of keeping the current direction when there is a choice between several cells with equal quantity of pheromone. Figure 7.a shows the regularity of the pheromone deposit (the gradient) and as a consequence the optimum paths followed by agents. This emergent solution is equivalent to the unique cycle strategy proposed in [1][3] for patrolling in a graph.

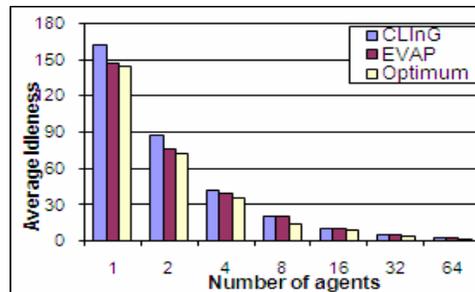
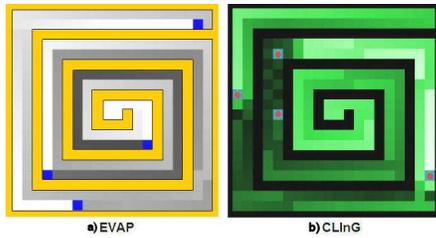


Figure 6. Spiral topology, average IGI

Gradients in Figure 7.b show the idleness propagations in CLInG (the highest values are the lightest). It appears that these gradients are not regular and may generate noise in agents search (cf. section 6).

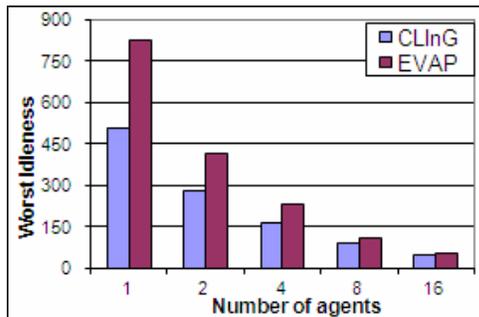


**Figure 7. Map B (screenshot)**  
 a) Pheromone Values of EVAP (max is light)  
 b) Propagated Idleness of CLInG (max is light)

### Environment with obstacle density (Map C)

We measured performances of both models with 4 agents in a 20x20 cells environment having 20% obstacles (agents are initially randomly located). Concerning the average idleness, both methods are close to the theoretical optimum value, with a slight advantage to CLInG algorithm. This advantage appears clearly in Figure 8 with the worst idleness measures. However this distance declines when the number of agents increases.

In the following sub-section, we consider more complex topologies in order to identify those creating the most important distance between the two methods.



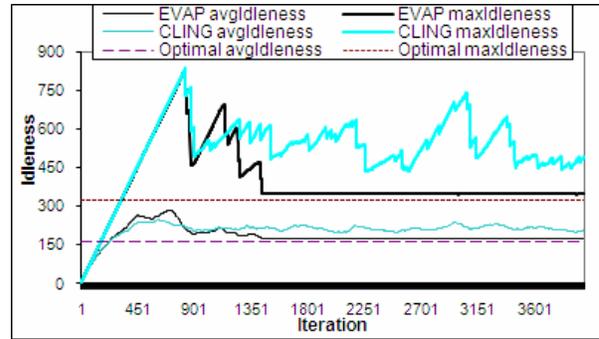
**Figure 8. 20% obstacles topology, average IWI**

### 5.3. Complex environments

#### Corridor – rooms (Map D)

We study now the behavior of both models towards complex environments composed of several rooms. We first start with map D topology (agents always start at the entrance of the corridor: the room at the bottom).

Figure 9 illustrates the performances in average and worst idleness on map D for the particular case of using a single agent. In fact, this case shows that the average and worst idleness of EVAP converges to the optimum performance at the same time. CLInG is a little less competitive and do not stabilize to a constant performance. Whenever we increase the population size, we note, however, that the performance of two methods are identical and remain close to the theoretical optimum values, without being stable.

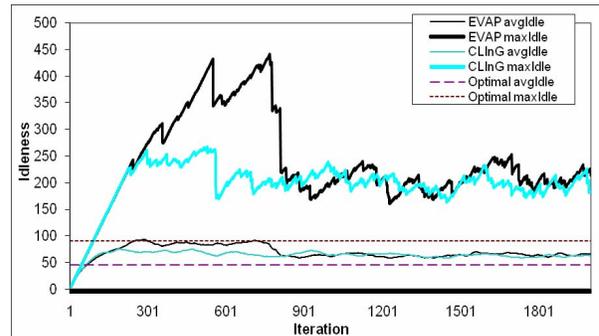


**Figure 9. Corridor-rooms topology, 1 agent, 4000 it**

Globally the simulations with this topology do not show a real difference of performance between both models. On the other hand, we show below that "imbricate" rooms constitute a topology discriminating the performance.

#### Imbricate rooms (Map E)

Figure 10 presents the performances in average and worst idleness of 4 agents in a 20x20 cells environment consisting of six imbricate rooms (cf. topology in Figure 11). The agents are initially all located at the same place (right bottom corner of the environment).



**Figure 10. 6-rooms topology, 4 agents, 2000 it.**

Over 2000 iterations, two distinct phases appears clearly for EVAP model. We observe in Figure 10, up to iteration 830, that the average idleness, as well as the worst idleness, is higher than CLInG values. This can be explained by the fact that agents have a first phase of exploration, which consists in getting into the rooms for the first time (cf. Figure 11.a).

Then, a second phase, which consists in re-visiting the rooms, is more efficient due to the fact that the pheromone leads the agents directly to the most distant rooms - the more formerly visited - (cf. Figure 11.b).

The difficulty for EVAP is situated at the doors separating two rooms. An agent who explores a room and arrives close to a door has an equal probability to continue the room exploration or to enter in the next room. If it chooses to continue the exploration, it risks to not pass again close the door and therefore to ignore

the unvisited room. We find here the problem identified by Wagner [11], due to the local vision of agents, of the choice between two nodes with equal interest (cf. details in section 3.3).

CLInG does not suffer this problem because the unexplored rooms propagate a strong idleness which ensures that agents, close to a door, will be attracted by the unvisited room. This propagation allows agents to get into rooms following an optimum way from the first exploration (cf. Figure 11.c).

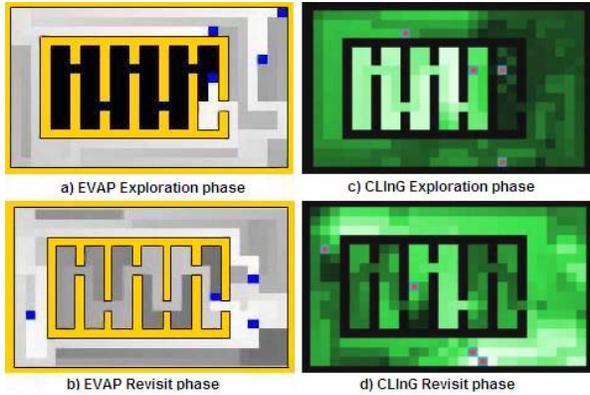


Figure 11. EVAP and CLInG, Map E (screenshot)

As we mentioned in section 3.3, the problem of the choice between several nodes, for the EVAP model, is reduced proportionately to the number of agents. The measures presented in Figure 12 verify this hypothesis (variation of agents number for an identical configuration).

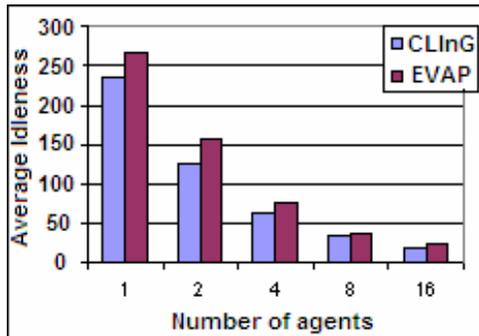


Figure 12. 6-rooms topology, average IGI

Concerning the EVAP model, it is interesting to focus on the exploration phase, which corresponds to the formation of a global information. Indeed, the order of rooms exploration, induced by the topology, build a gradient of pheromone which is necessarily oriented following this order. So, once the gradient is formed, an agent will get into the rooms efficiently while generating systematically the inverse gradient. That allows it to alternate getting in and getting out in an optimum way. This explains the system converges to a stable performance.

## 6. Discussion

### 6.1. Model complexity

The objective of the previous section was to measure the interest of introducing the propagation of information compared to a model which only uses the evaporation process. Indeed, CLInG can be defined as the algorithm EVAP (idleness playing the role of the pheromone) augmented with the diffusion of information through the environment.

We identified topologies where CLInG proves to be more effective due to this propagation of information. However this process has a cost. The difference of algorithms complexity is in the calculation performed by environment at each iteration. More precisely, in EVAP, for  $c$  cells containing the pheromone, it takes  $c$  evaporation operations (eq. 1). Thus for a  $n \times n$  cells environment, it takes at most  $n^2$  evaporation operations. The environment of CLInG is much costly since it needs  $n^2$  idleness calculation operations plus  $n^2$  spread operations. In practice, CLInG proves to be twice as costly in execution times.

### 6.2. Exploration and patrol

The simulations carried out on complex environments (corridor and imbricate rooms) revealed two functioning phases, in particular for EVAP. Initially, the system performs a first exploration of the environment, then, it changes brutally for more stable and effective behaviour.

These two phases also exist for CLInG but, generally, the first phase is shorter than the EVAP one. It is due to attractions induced by the not yet explored areas (see Figure 9 and Figure 10). It is interesting to pay attention that in both methods the system self-organizes via the marking of the environment, and converges towards an effective patrolling (possibly optimal, see EVAP on map B).

### 6.3. Advantages and drawbacks of models

EVAP is an effective solution for average complexity environments (non-obstacle, spiral, corridor-rooms). We have shown that the deposit and the evaporation of a pheromone guarantee a low average idleness.

On the other hand, considering the worst idleness, CLInG proves generally to be more efficient than EVAP, except for spiral and corridor topologies when the number of agents is low.

More generally, a surprise of this study is the good performances, even optimal, for mono-agent patrolling. It shows that marking the environment can constitute a

good solution for mono-agent problems, while guaranteeing scalability on the agent number. Indeed, the average idleness decreases linearly with the number of agents, and makes it possible to be more efficient on the exploration phase.

The study has shown that CLInG is more efficient than EVAP on complex environments composed of imbricate rooms, particularly if the number of agents is low. The difference of performances reduces gradually when the number of agents increases. This is the consequence of using a swarm approach (see Figure. 12 and section 3.3).

The information propagated in CLInG model naturally generates complexity in the system. While attracting agents towards zones of interest, this process can also generate noise, for example by attracting several agents to the same cell. It explains why CLInG does not reach to stabilize to a stationary global behavior (as observed on the spiral topology).

## 7. Conclusion

In this article, we have tackled the multi-agent patrolling problem in unknown environments. Algorithm CLING [9], one of the rare propositions, exploits the marking of the environment and the diffusion of information. We proposed a simpler model (EVAP) exploiting only the evaporation of a pheromone (no diffusion). The experimental study on environments with growing complexity has shown that the propagation of information is efficient only with some environment topologies. For this reason, we have identified the N-imbricate-rooms problem which defines a new case study for this problem.

We have shown that EVAP model is less efficient for the initial exploration phase but then converges to an average performance identical to CLInG. This difference of performance is reduced when the population increases, showing the collective nature of the proposed models. So, the propagation process is proved to be an accelerator of the exploration phase of the multi-agent patrolling. Moreover, this experimental study has indicated that these algorithms could constitute a competitive solution for the mono-agent patrolling problem.

We plan as perspectives of this work to go deep in the study of CLInG parameters' role, to add energy limitation and dynamic obstacles variants. Finally, we will implement EVAP in a multi-robot simulator to evaluate the model on more realistic environments.

## References

[1] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Meuzes, V. Corruble, Y.

- Chevaleyre et al. "Recent Advances on Multi-Agent Patrolling", *Proceedings of the 17th Brazilian Symposium on Artificial Intelligence*, pp. 474-483, 2004.
- [2] E. Bonabeau, M. Dorigo, G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999
- [3] Y. Chevaleyre, "The Patrolling Problem", *Technical Report*, Paris 9, [http://www.lamsade.dauphine.fr/~chevaley/papers/anna\\_patro.pdf](http://www.lamsade.dauphine.fr/~chevaley/papers/anna_patro.pdf), 2003.
- [4] F. Lauri, F. Charpillet, "Ant Colony Optimization applied to the Multi-Agent Patrolling Problem", *IEEE Swarm Intelligence Symposium*, 2006.
- [5] A. Machado, G. Ramalho, J.-D. Zucker, A. Drogoul, "Multi-Agent Patrolling: an Empirical Analysis of Alternative Architectures", *Proceedings of Multi-Agent Based Simulation*, pp. 155-170, 2002.
- [6] L. Panait, S. Luke, "A pheromone-based utility model for collaborative foraging", *Proceedings AAMAS*, pp. 36-43, 2004.
- [7] H.V. Parunak, M. Purcell, R. O'Connell, "Digital Pheromones for Autonomous coordination of swarming UAV's", *Proceedings of AIAA First Technical Conference and Workshop on Unmanned Aerospace Vehicles, Systems, and Operations*, 2002.
- [8] H.Santana, G.Ramalho, V. Corruble, R. Bohdana. "Multi-Agent Patrolling with Reinforcement Learning", *Proceedings of the 3<sup>rd</sup> international Joint Conference on Autonomous Agents and Multi-Agent Systems*, pp. 1122-1129, 2004.
- [9] F. Sempé, "Auto-organisation d'une collectivité de robots: application à l'activité de patrouille en présence de perturbations", *PhD Thesis*, Université Paris VI., 2004
- [10] F. Sempé, A. Drogoul, "Adaptive Patrol for a Group of Robots", *Proceedings of the 2003 IEEE/RSJ, Intelligence Robots and Systems*, Las Vegas, Nevada, pp. 2865-2869, 2003.
- [11] I.A. Wagner, M. Lindenbaum, A.M. Bruckstein, "Distributed Covering by Ant-Robots using Evaporating Traces", *IEEE Transactions on Robotics and Automation*, vol.15, n.5, pp. 918-933, 1999.
- [12] V. Yanovski, I.A. Wagner, A.M. Bruckstein, "A Distributed Ant Algorithm for Efficiently Patrolling a Network", *Algorithmica*, vol.73, pp. 165-186, 2003.