

Multi-agent Patrolling: Multi-Objective Approach of the Event Detection by a Mobile Wireless Sensors Network

Elie Tagne Fute
University of Dschang
International Research Center for Application
of Space Technologies (CIRATS)

Fabrice Lauri
Systems and Transportation Laboratory
University of Technology of Belfort-Montbéliard

Emmanuel Tonye
Electronic and Signal Treatment Laboratory
University of Yaounde I

Abderrafaa Koukam
Systems and Transportation Laboratory
University of Technology of Belfort-Montbéliard

ABSTRACT

Patrolling an environment involves a team of agents whose goal usually consists in continuously visiting its most relevant areas as frequently as possible. Hence, patrolling consists in continuously visiting relevant areas of an environment, in order to efficiently supervise or control it. The Ant Colony Optimization (ACO) is adopted as the solution approach that efficiently solves the multi-agent patrolling problem. Two strategies are employed: all agents are located on the same node at the initial time, and the agents are dispersed over the graph, they start to patrol from their new locations. This paper mainly describes the formulation problem of event detection by a multi-objective approach, an ACO and multi-agent approach are used to model and simulate the detection of events. Different parameters are evaluated in order to minimize them. The minimization of Worst Idleness, Energy consumption and Communicational Idleness are not compatible. It is therefore necessary to seek compromise solutions. The set of compromise solutions is called Pareto Front. The set of solutions and Pareto Front are presented respectively for the cases of non-dispersion and dispersion of agents.

General Terms:

Ant Colony Optimization, compromise solution, multi-agent patrolling problem, mobile wireless sensors

Keywords:

Agent, ant, mobile sensors, optimization, patrolling

1. DESCRIPTION OF THE PROBLEM

Mobile wireless sensors are used here to ensure a dynamic coverage of an environment in order to detect and react as fast as possible to some unpredictable and dynamic phenomena such as fire, flood and pollution for example.

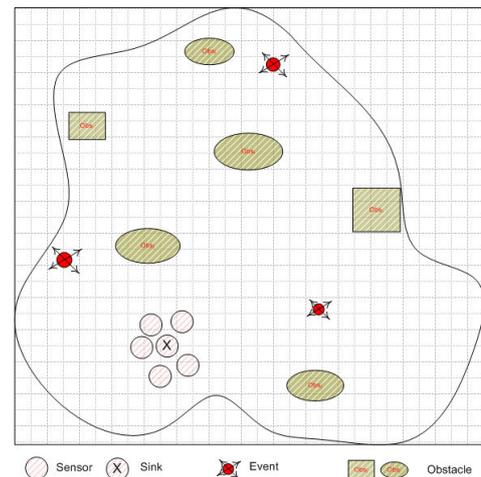


Fig. 1. A typical environment to patrol.

1.1 Environment Description

Figure 1 presents an example of an environment where a network of mobile wireless sensors may patrol. Mobile sensors are deployed in the environment at a given area. They will have to move with coordination in order to avoid obstacles and to minimize the delay between the detection of a phenomenon and the reaction to this phenomenon.

Coordination can be ensured by one or several sink nodes. Sink nodes are heavily equipped sensors that gather informations from other sensors and communicate these informations to a base station. The base station merges the received informations and can send new orders to the sink. The sink node can then transmit these orders to the appropriate sensors.

Detecting phenomena that emerge unpredictably in an environment can be managed efficiently by performing a patrolling task. Patrolling using a mobile wireless sensor network can be divided into two stages. At first, sensor nodes are deployed in the environment at a given location. Then, they elaborate a formation around a sink

node and then start to patrol in the search of critical areas, by maintaining the formation. Indeed, a formation ensures that all of the mobile sensors will be able to communicate with each other and with the sink if it is necessary, so that the base station will receive up-to-date information as frequently as possible to react appropriately. Once a critical phenomenon has been detected by a sensor node, it sends a warning to its sink node. The sink node transmits it to the base station that can finally warn other specialized agents, such as firefighters to handle wildfires for example. To better help the specialized agents, the patrolling sensors may have received orders forcing them to reorganize themselves to follow the phenomenon expansion.

1.2 Mobile Sensor Model

A mobile sensor node is made up of several components: a *sensing unit*, a *processing unit*, a *transceiver unit*, a *power unit*, a *location finding system* and a *mobilizer*. Sensing units are usually made of sensors (e.g. thermal, visual, infrared, radio...) and analog to digital converters (ADCs). The analog signals perceived by the sensors are converted to digital signals by the ADCs. Digital signals are then used by the processing unit to make the sensor node collaborate with the other sensor nodes to fulfill the assigned tasks. The transceiver unit (e.g. a radio frequency device) enables a sensor to communicate with other nodes in the network. The power unit supplies current to the other sensor components [6]. It is one of the most important units since the lifetime of a sensor network depends on the lifetime of the power resources of each sensor node. Most of the sensor network perception tasks and routing techniques require the knowledge of location with high accuracy, provided by the location finding system.

Each sensor node is characterized by an occupation radius r_{occ} that represents the surface it takes on the ground. It can perceive obstacles and other nodes within its perception radius r_{per} and it can communicate with the other nodes located within its communication radius r_{com} , with $r_{com} > r_{per} > r_{occ}$.

2. RELATED WORKS

Firstly, the mathematical framework of the patrolling task is presented. The multi-agent patrolling problem is usually specified formally as in [12, 3, 11, 7, 6]. The environment to patrol is reduced to a graph $G = (V, E)$. V stands for the convex areas and E the roads between them. A cost $c(i, j)$, associated with each edge (i, j) , measures the time required to reach the node j from the node i .

Let sensor agents bound to visit the areas defined in the graph G at regular intervals. Each agent is located at one of the nodes of V at the initial time.

Solving the patrolling problem consists of elaborating a multi-agent graph strategy π . Such a strategy must optimize one or several given quality criteria. $\pi = \{\pi_1 \cdots \pi_n\}$ is made up of the individual strategies π_i of each agent i . An individual strategy π_i is defined such that $\pi_i : \mathbb{N} \rightarrow V$, $\pi_i(j)$ denoting the j -th node visited by the agent i , with $\pi_i(j+1) = x$ if $(\pi_i(j), x) \in E$.

It is commonly admitted that a relevant patrolling strategy is one that minimizes, for each node, the time spent between two visits to the same node.

Several criteria have been devised in [12, 14, 7] in order to evaluate the quality of a π after T time steps (or *cycles*) of simulation. All of them are based on the notion of (INI).

The INI $I_t^\pi(i)$ of a node i at time t following the π is the number of time steps this node remained unvisited. By convention, at the initial instant, $I_0^\pi(i) = 0, \forall i = 1, 2, \dots$, and for any π .

At a given instant t , the WI_t^π is the highest INI encountered since t time steps of simulation when following the π , i.e.:

$$WI_t^\pi = \max_{i \in V} I_t^\pi(i)$$

A π can be evaluated after T cycles of simulation using the WI_T^π . The is the highest INI observed during the T -time steps of the simulation, given by the following equation:

$$WI_T^\pi = \max_{t=1,2,\dots,T} \max_{i \in V} I_t^\pi(i)$$

In [12, 3, 4, 11, 9], authors use agents to do the patrolling task and the evaluation of patrolling strategies is based on WI. They don't consider the constraints relative to the sensor network, e.g energy consumption or communications between sensors with the aim to send to the sink the events detected.

In [7, 14], agents are wireless sensor networks but, evaluation of patrolling strategies is based on the worst idleness.

[6] describes the design of a system for wildfire monitoring incorporating wireless sensors. The environment to patrol will be represented by a regular grid. In the modelling of the evaluation criterion, a multi-agent patrolling strategy should minimize the worst idleness criteria and the energy consumed by the network.

3. MATHEMATICAL FRAMEWORK OF THE PHENOMENON DETECTION TASK

We define the mathematical framework of the phenomenon detection task following three components namely the criterion of the Graph Idleness (WI), the criterion of Energy consumption of the network (E) and the criterion of Communicational Idleness (CI).

3.1 Criterion of the graph idleness

The classic problem of multi-agent patrolling is to determine a strategy s that minimizes the value of the worst idleness $WI(G)$ or that of the average idleness $AI(G)$ [14, 7]. The worst idleness highlights the greatest time passing on all nodes in the environment unlike the average idleness that favours the sum of the times on all the nodes. Thus, minimizing the worst idleness will result to minimize the detection or perception time of an event on the whole of the graph, while in the minimization of the average idleness, the difference between the smallest and largest idleness can be significant and may lead to greater idleness at certain nodes. Therefore, an assessment based on the minimization of the worst idleness of the graph ($\min_s(W(G))$) seems more objective than that based on the minimization of the average idleness of the graph ($\min_s(AI(G))$). Moreover, the value of $W(G)$ is considered equal to the largest interval of time during which a cell is visited by no agent.

3.2 Criterion of energy consumption of the network

In the *multi-sensor patrolling* problem, agents are mobile wireless sensors, therefore we must take into account issues related to their energy consumption. The total energy consumed by a sensor i at time t when following a π is dependent on the energy consumed by its computation system, its communication system, its perception system, its locomotion system and its inactivity system between times $t-1$ and t .

The energy consumed by a sensor i for a patrolling strategy π between times $t-1$ and t can be expressed by the equation (1):

$$E_i^\pi(t) = E_{com_i}^\pi(t) + E_{comp_i}^\pi(t) + E_{mov_i}^\pi(t) + E_{percept_i}^\pi(t) + E_{idle_i}^\pi(t) \quad (1)$$

where

$E_{com_i}^\pi(t)$, $E_{comp_i}^\pi(t)$, $E_{mov_i}^\pi(t)$, $E_{percept_i}^\pi(t)$ et $E_{idle_i}^\pi(t)$ represent the energy of communication, computation, movement, perception and inactivity of the sensor i for the strategy π between times $t-1$ and t . These energies are expressed in Watt-hour.

In [13], the energy expression is proportional to the amount of traffic passing through each sensor. The expression of the energy considered takes into account of the local traffics at each sensor, which enables an assessment of the energy consumed at the level of each sensor. Thus, expression (2) allows the evaluation of the energy consumed in communication of sensor i for a patrolling strategy π between times $t-1$ and t .

$$E_{com_i}^\pi(t) = e_i^{rec} \times \sum_{j \in v_i^t} p_{j,i}^t + e_i^{env} \times p_i^t \quad (2)$$

where

e_i^{rec} represents the energy consumed by the sensor i due to the reception of a packet. e_i^{env} represents the energy consumed by the sensor i due to the transmission of a packet. $p_{j,i}^t$ represents the number of packets relayed/transmitted by the sensor i from the sensors in direct communication with i and this between the instants $t-1$ and t . p_i^t represents the number of packets going from sensor i (sensor i is considered as the source) towards those in direct communication with i and this between the instants $t-1$ and t . v_i^t represents the set of sensors in direct communication with the sensor i between the instants $t-1$ and t .

Expression (3) below enables the evaluation of the energy consumed in terms of calculation of the sensor i in a patrolling strategy π between the instants $t-1$ and t .

$$E_{comp_i}^\pi(t) = n_{opa}^{i,t} \times e_{opa} + n_{opp}^{i,t} \times e_{opp} \quad (3)$$

where

$n_{opa}^{i,t}$ and $n_{opp}^{i,t}$ represent respectively the number of operations of addition/subtraction and multiplication/division performed by the sensor i between the instants $t-1$ and t .

e_{opa} and e_{opp} represent the energies required to perform respectively an operation of addition/subtraction and multiplication/division.

Since sensors are mobile, they are endowed with engines. We express the evaluation of the displacement energy by expression (4). In this expression, $w_i^{j,t}$ indicates for a patrolling strategy π the average labour done by the engine j of sensor i between $t-1$ and t . M_i is the number of engines of sensor i .

$$E_{mov_i}^\pi(t) = \sum_{j=1}^{M_i} w_i^{j,t} \quad (4)$$

Expression (5) expresses for a patrolling strategy π , the evaluation of the perception energy of sensor i . k_i^t represents the number of perceptions performed by sensor i between the instants $t-1$ and t . $e_i^{percept/m}$ represents the energy consumed by sensor i when the radius of perception increments of a unit of measure (one meter). R_i represents the perception radius of the sensor i .

$$E_{percept_i}^\pi(t) = k_i^t \times e_i^{percept/m} \times R_i \quad (5)$$

Expression (6) expresses for a patrolling strategy π , the evaluation of the energy consumed by the sensor i during its inactivity between $t-1$ and t . When the sensor is not in communication, it performs no calculation, no movement and no perception, it can still adjust or

perform rotation of its arms and other devices, it can also reorganize its internal data. e_{ajust}^t and e_{reorg}^t represent the energy consumed associated respectively to the execution of any of the tasks mentioned previously and α and β are multiplying coefficients.

$$E_{idle_i}^\pi(t) = \alpha e_{ajust}^t + \beta e_{reorg}^t \quad (6)$$

One of the objectives to achieve is to be able to minimize the sum of the energies consumed by the entire network. For a patrolling strategy π , we express the energy consumed by the network by the expression E^π below.

$$E^\pi = \sum_{t=0}^T \left(\sum_{i=1}^{N_{sens}} E_i^\pi(t) \right) \quad (7)$$

Expression (7) above evaluates the total energy consumed by the entire network throughout the simulation period T by the patrolling strategy π .

3.3 Criterion of communicational idleness

The maximum delay of communication between a sink node and its base station can be defined by using the notion of *communicational idleness* at the level of a sensor node. Here, the communicational idleness of a sensor node represents the number of time steps the sensor node has not been in communication with a sink. A sink and a sensor node n can communicate with each other either when they are found within the same communicational radius or when several sensor nodes including the sink and the sensor node n constitute a communicational chain.

Let's define the maximum communication delay between the Sink and the other sensors in a group by the notion of *communicational idleness*. CI of a sensor is the number of time steps spent since the last communication between this sensor and the Sink. It should be noted that the communication between the Sink and the other sensors of the group can be direct or indirect. When direct, the Sink is in the sensor's communicative field. An indirect communication means the communication between a sensor and Sink is realized by intermediate sensors forming a communicational channel.

Let $CI_i^\pi(t)$ the communicational idleness of the node i at time t for patrolling strategy π . This value remains 0 for each sensor that is not the Sink as long as no event occurs. It is incremented at the onset of an event. Assuming that j is the Sink, the maximum communication delay between the Sink and the other sensors is defined by:

$$CI_T^\pi = \max_{t=0}^T (\max_{i \in S_C} CI_i^\pi(t)) \quad (8)$$

where

T represents the total simulation time, but can also represent the time after which the Sink reaches its threshold energy. $S_C = \{1, 2, \dots, j-1, j+1, \dots, N_{sens}\}$.

N_{sens} represents the number of sensor in the network.

3.4 Synthesis and overall formulation of the problem

The phenomenon detection task by mobile sensors is a multi-objective problem, that consists of seeking for a *Pareto optimal* multi-agent strategy π^* .

The problem of multi-sensor patrolling aims to determine a patrolling strategy which optimizes the following criteria :

$$\begin{cases} I(\pi) = \min_{\pi} \left(\max_{t \in [0, T]} (WI_t^{\pi}(G)) \right) \\ \quad = \min_{\pi} \left(\max_{t \in [0, T]} (\max_{i=1, \dots, N} (I_i^{\pi}(t))) \right) \\ CI(\pi) = \min_{\pi} \left(\max_{t=0}^T (\max_{i \in S_C} CI_i^{\pi}(t)) \right) \\ E(\pi) = \min_{\pi} \left(\sum_{t=0}^T \left(\sum_{i=1}^{N_{sensors}} E_i^{\pi}(t) \right) \right) \end{cases}$$

where

$I(\pi)$ represents the minimization of the worst idleness of the graph's nodes or the cells the environment.

$CI(\pi)$ represents the minimization of the maximum communication delay between the sensors and the Sink at the appearance of an event.

$E(\pi)$ represents the minimization of the total energy consumed by the entire network.

π is the patrolling strategy searched.

T is the total simulation time, also representing the time after which the Sink has reached its threshold energy.

N represents the number of nodes or cells of the environment.

$N_{sensors}$ represents the number of sensors of the network.

$S_C = \{1, 2, \dots, j-1, j+1, \dots, N_{sensors}\}$.

3.5 Multiobjective Optimization

Multi-objective optimization, also known as multi-criteria or multi-attribute optimization, is the process of simultaneously optimizing two or more conflicting objectives subject to certain constraints. A Multi-objective Problem (MOP) includes a set of n parameters (decision variables), a set of m objective functions and a set of k constraints [15, 10]. Multi-objective optimization seeks to optimize several components of a cost function vector [5, 8, 1, 2].

The main goal in the resolution of MOP is to obtain the set of compromise solutions called the Pareto optimal set or the Pareto Front. Globally, we have two phases in solving the MOP. In the first phase, we have the determination of the Pareto optimal set. In the second phase, we have the choice of a solution according to the preferences of the decision maker. This choice requires knowledge about the treated problem and the factors related to it. Thus, a solution chosen by a decision maker could be not be acceptable for another decision.

A MOP can be defined as follows :

$$\text{minimize}(f(x)) = \{f_1(x), f_2(x), \dots, f_m(x)\}$$

with

— $x \in S$,

— $m \geq 2$ represents the number of functions to optimize,

— $x = \{f_1(x), f_2(x), \dots, f_n(x)\}$ represents the decision vector,

— S represents the set of the realizable solutions,

— $f(x)$ is the vector of the criteria to be optimized.

Some concepts are needed for determining the Pareto front and preference solutions.

Definition 1: A solution $y = \{y_1, y_2, \dots, y_n\}$ dominates a solution $s = \{s_1, s_2, \dots, s_n\}$ if and only if $\forall i \in [1..n], s_i \leq y_i$ and $\exists j \in [1..n] / s_j < y_j$.

If a solution x dominates a solution y then we can say that x is a better solution than y . The solutions are non-dominated if none of the solutions dominates the other.

Definition 2: A solution $x^* \in S$ is Pareto optimal if and only if there does not exist a solution $x \in S$, such as $f(x)$ dominates $f(x^*)$.

Definition 3: The Ideal point $z^* = (z_1^*, z_2^*, \dots, z_n^*)$ is the point which optimizes each objective function f_i , thus we have :

$$z_i^* = \min (f_i(x)), x \in S.$$

Definition 4: The Nadir point $z^{nad} = (z_1^{nad}, z_2^{nad}, \dots, z_n^{nad})$ is the point which maximizes each objective function f_i , thus we have : $z_i^{nad} = \max (f_i(x)), x \in S$.

The definitions 3 and 4 are needed for determining the solutions according to the preferences of the decision maker.

In order to select the best solution in the set of solutions, we define the distance between a potential solution and an Ideal Point. A first objective is to provide an approach for choosing a compromise solution in a set of solutions. Another objective is to provide compromise solutions automatically, because we would not want the user to do it manually.

Let us consider:

$$\text{Ideal Point: } P^* = \{P_1^*, P_2^*, \dots, P_{N_{obj}}^*\}$$

$$\text{Nadir Point: } P^{nad} = \{P_1^{nad}, P_2^{nad}, \dots, P_{N_{obj}}^{nad}\}$$

where

— N_{obj} represents the number of objective functions,

— P_i^* represents the $\min_{s \in S} f_i(s)$

— P_i^{nad} represents the $\max_{s \in S} f_i(s)$

— $f_i(s)$ is the i^{th} objective function

Using the normalized distance between ideal point and potential solution k given by $P^k = \{P_1^k, P_2^k, \dots, P_{N_{obj}}^k\}$, the compromise programming(CP) method allows to determine the solutions to the ideal point in representing different objectives within a single function:

$$\text{dist} = \sqrt{\sum_{i=1}^{N_{obj}} p_i \left(\frac{P_i^k - P_i^*}{P_i^{nad} - P_i^*} \right)^2}$$

where p_i represents the preferences for objective functions, such that $\sum_{i=1}^{N_{obj}} p_i = 1$ with $p_i \geq 0$ and $P^k = \{WI_T^{\pi}, CI_T^{\pi}, E_T^{\pi}\}$

4. ACO APPROACH TO THE PATROLLING PROBLEM

4.1 Structure of the ACO algorithm used for sensor patrolling

The Ant Colony Optimization (ACO) is adopted as the solution approach that efficiently solves the multi-agent patrolling problem. Two strategies are employed. At first, all the agents are located on the same node at the initial time. Afterward, the agents are dispersed over the graph, they start to patrol from their new locations. We propose an ACO algorithm that determines the multi-agent patrolling strategy by using or not a dispersion of the agents sensors. For the Multi-Agents Patrolling Problem (MAPP), each of the r patrolling agents i wants to find an individual strategy π_i (the list of nodes it has to visit) such that the multi-agent patrolling strategy $\pi = \{\pi_1, \pi_2, \dots, \pi_r\}$ optimizes a given quality measure. Using the metaphor of ants working for a salesman and supposing m ant colonies will be used, then m ants (one ant of each colony) will work for one agent. Hence, in our approach, each ant elaborates an individual strategy and each colony builds a solution standing for a multi-agent patrolling strategy.

Inside a colony, an ant goes from one unvisited node to another adjacent unvisited node. Each ant k of the colony l records the list of the nodes it has already visited in $tabu_{k,l}$, and knows the already visited nodes of every other ant of the same colony. An ant selects

the next node to visit according to the probability $p_{ij}^{k,l}$. The probability of selection of a node by the ant k of the colony l is given by:

$$p_{ij}^{k,l} = \begin{cases} \frac{[\tau_{ij}(T)]^\alpha [\eta_{ij}]^\beta}{\sum_{u \in allowed_l} [\tau_{iu}(T)]^\alpha [\eta_{iu}]^\beta} & \text{if } j \in allowed_l \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where

V is the set of graph nodes, $allowed_l = \{V - \sum_{i=1}^r tabu_{i,l}\}$ is the set of the unvisited nodes of colony l , $\tau_{ij}(T)$ is the pheromone intensity on edge (i, j) at cycle T , $\eta_{ij} = 1/c_{ij}$ is the visibility of node j from node i and α and β are parameters that control the relative importance of pheromone intensity and visibility.

$$\tau_{ij}(T+1) = \rho \tau_{ij}(T) + \Delta\tau_{ij} \quad (10)$$

where

ρ is the evaporation coefficient, $\tau_{ij}(T+1)$ and $\tau_{ij}(T)$ are the pheromone intensities on edge (i, j) at cycle $T+1$ and cycle T , respectively. $\Delta\tau_{ij}$ is the pheromone quantity deposited on the edge (i, j) by all the ants of all the colonies in this cycle.

Penalty concept

In order to focus on strategies that promote communicative aspect, we introduced the concept of *penalty* (see Definition 5). The aim is to take into account the distance of a sensor agent from its *Sink*. In the simulation, the main matter is to operate on the score (see Definition 6) of a colony every time an ant of the colony departs from the *Sink*, that is whenever an ant is out of range of its *Sink*.

Definition 5: Penalty A penalty is a strictly positive real constant that is added to the score (see Definition 6) of a colony, in order to reduce its probability of selection as the best compared to other colonies.

Definition 6: Colony score The score of a colony is a cumulative sum of its worst environmental idleness and penalties.

4.2 Resolution approach by ACO

The evaluation of the different objective functions is performed by the *ColonyFitness(colony, idleness, penalty)* function on line 29 of algorithm 2. A detailed description of this function is given by algorithm 1. The function *ColonyFitness(colony, idleness, penalty)* returns as a result the worst idleness (L_c) of the colony taken into parameter, as well as the potential *Sink*. This worst idleness is necessary to assess the rate of pheromone deposited on an edge. In addition to returning the value of the worst idleness, algorithm 1 enables the evaluation and save of the three following functions : *worst idleness*(W^π), *communicational idleness*(CI^π) and *consumed energy*(E^π) for a patrolling strategy π .

We determine for each colony, the worst idleness of the graph. For each colony, we fix at each time each ant in the colony as *Sink*. Each ant is like a sensor agent with a communicative radius. In each case, the list of agents in direct and indirect communications is evaluated with the potential *Sink*. Based on this list, we can determine the list PL_i of the agents having no communication with the *Sink* agent i . It is then possible to evaluate the value of the penalty associated with the underlying strategy. As shown in algorithm 1, this value is proportional to the size of the list PL_i . The final score to consider

for the colony is the smallest value in the list of scores after adding penalties. Since these scores are evaluated for each of the colonies, the strategy that we shall select, will be the one with the smallest value in the list of scores. A value that corresponds to a strategy path of the nodes of the graph, well as an agent able to play the role of *Sink*.

The multi-agent patrolling strategies are applied to the sensor network. We propose an agent who will play the *sink* role. The *sink* must communicate with all the rest of sensors.

We present in Table 1, the values that we used in the main algorithm for the parameters *number of iterations*, *number of colonies*, *number of ants by colony* and the values of *penalty*.

Table 1. Parameters of the simulation

Parameters	Values
Number of iterations (T)	50000
Number of colonies	1,2,5,10
Number of ants per colony	1,2,3,4,5,10,15
Penalty	5.0, 10.0 → with penalty; 0.0 → without penalty

Algorithm 1 : Algorithm for the computation of $L_l(\text{colony}, WI_c, \text{penalty})$

- 1: Compute WI_{L_l} as being the worst idleness of the graph after P time steps of simulation.
- 2: **for** each ant i of the colony l **do**
- 3: $Score[i] \leftarrow WI_c$
- 4: Evaluate the list of agents in direct and indirect communication with the *Sink*= i
- 5: $Com_i \leftarrow ListAgentsCommunication(i)$
- 6: Evaluate the energy consumption of the group with *Sink*= i
- 7: $E_i \leftarrow EnergieConsoGroup(i)$
- 8: Evaluate the best worst communicational idleness between *Sink* i and other agent-sensor f ($f \neq i$)
- 9: $CI_i \leftarrow OisiveteCom(i)$
- 10: $PL_i \leftarrow$ ants list having no communication with i : $L - Com_i$;
- 11: $CScore[i] \leftarrow CScore[i] + size(PL_i) \times penalty$;
- 12: **end for**
- 13: $Score[c] \leftarrow$ smaller value of $CScore[]$;
- 14: $Sink[c] \leftarrow$ ant's ID minimizing $Score[]$;
- 15: Determine the solutions with the smallest energy(E_s) and corresponding value of CI_s where s is the *Sink* selected.
- 16: Save the values of WI_s , E_s and CI_s
- 17: Return $Score \cup Sink$
- 18: Stop

Algorithm 2 ACO algorithm for multi-agent patrolling

Require: G : graph, N : list of initial nodes of $r = |N|$ agents,
 $penalty$: penalty value, n : colony number, T_{MAX} : number
of maximum iteration of the algorithm,

Ensure: $result$: Set of compromise solutions.

```

1:  $T \leftarrow 0$  ( $T$  is the cycles counter)
2: for every edge  $(i, j)$  of  $G$  do
3:   Initialization of the phomone quantities  $\tau_{ij}(T) = c$  (or
    $\tau_{ij}^k(T) = c, \forall k$ )
4:   Initialize  $\Delta\tau_{ij} = 0$  (or  $\Delta\tau_{ij}^k = 0, \forall k$ ).
5: end for
6: Place the  $r$  ants of each colony on the starting node of the cor-
   responding agents.
7: while  $T < T_{MAX}$  do
8:   for each colony  $l$  do
9:     for every ant  $k$  do
10:      Reset  $tabu_{k,l}$  and place the starting node of ant  $k$  in
       $tabu_{k,l}(0)$ .
11:    end for
12:  end for
13:  for each colony  $l$  do
14:     $s \leftarrow 0$ 
15:    repeat
16:       $s \leftarrow s + 1$ 
17:      for each ant  $k$  do
18:        Choose the node  $j$  to move to, with probability  $p_{ij}^{k,l}$ 
        given by equation (9)
19:        Move the ant  $k$  at node  $j$  and insert node  $j$  in
         $tabu_{k,l}(s)$ 
20:      end for
21:      until  $\cup_{i \in [1;r]} tabu_{i,l} = V$ 
22:      Evaluate  $WI_l$  as the worst idleness of the graph associ-
      ated with the colony  $l$ .
23:    end for
24:    for each colony  $l$  do
25:      for each ant  $k$  do
26:        Place the ant  $k$  on the node indicated by  $tabu_{k,l}(0)$ 
27:        Build in  $walk_{k,l}$  the closed-walk in  $G$  using the paths
        computed with  $A^*$ .
28:      end for
29:       $result \leftarrow ColonyFitness(c, WI_c, penalty)$ 
30:       $L_c \leftarrow result.Score(c)$ 
31:       $Sink[c] \leftarrow result.Sink(c)$ 
32:    end for
33:    Determine and save the minimum value of  $L_c[]$  and
     $Sink(Sink[])$  corresponding;
34:    for every edge  $(i, j)$  of  $G$  do
35:      for every ant  $k$  of the best colony do
36:         $\sigma_{ij}^{k,l} = \begin{cases} \frac{Q}{L_l} & \text{if } (i, j) \in walk_{k,l} \\ 0 & \text{otherwise} \end{cases}$ 
37:         $\Delta\tau_{ij} = \Delta\tau_{ij} + \sigma_{ij}^{k,l}$ 
38:      end for
39:    end for
40:    for every edge  $(i, j)$  of  $G$  do
41:      Compute  $\tau_{ij}(T+1)$  (or  $\tau_{ij}^k(T+1), \forall k$ ), according to
      equation (10).
42:    end for
43:     $T \leftarrow T + 1$ 
44:    for every edge  $(i, j)$  of  $G$  do
45:      Set  $\Delta\tau_{ij} = 0$  (or  $\Delta\tau_{ij}^k = 0, \forall k$ ).
46:    end for
47:    Reset all the lists and scores
48:  end while

```

5. EXPERIMENTAL RESULTS

The experiments were performed on six maps, which are shown on figure 2. These maps present the set of environments with different complexities where we did the simulations.

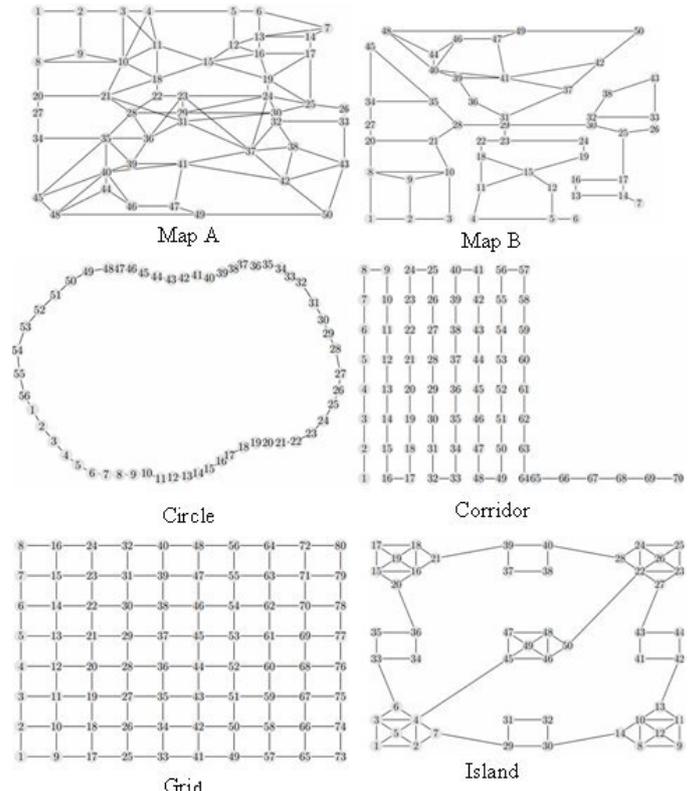


Fig. 2. Maps with different corresponding graph topologies on our simulator

Multi-agent patrolling strategies were computed on six different graph topologies of various complexities (table 2). This table presents according to the Map, its name, number of nodes, edges and its degree.

Table 2. Graph topologies

Name	# nodes	# edges	Degree
Circle	56	56	2
Corridor	70	69	2
Map B	50	69	5
Grid	80	142	4
Island	50	84	6
Map A	50	106	7

We evaluated the average number of meetings ($AvMeet$), and the number potential solutions ($NbSol$) depending on the various topologies for the respective cases of five and ten sensors agents.

Implementation without dispersing agents across the graph

The tables 3 and 4 present the results where we supposed no

dispersion of agents. On these tables, *SP* and *AP* indicate the data in the cases respectively *Without penalty* and then *With penalty*.

Table 3. Case without dispersion: meetings with five sensors agents

Name	AvMeet(SP)	#NbSol(SP)	AvMeet(AP)	#NbSol(SP)
Map A	20.0	21	21.28	21
Map B	27.22	27	36.85	20
Island	17.26	45	23.23	26
Circle	56.42	21	58.93	47
Corridor	153.62	32	204.33	24
Grid	30.89	28	47.75	24

Table 4. Case without dispersion: meetings with ten sensors agents

Name	AvMeet(SP)	#NbSol(SP)	AvMeet(AP)	#NbSol(SP)
Map A	7.07	42	8.14	21
Map B	17.79	24	18.0	32
Island	8.11	18	8.42	35
Circle	21.88	35	29.95	20
Corridor	83.30	23	92.63	44
Grid	13.95	22	15.81	32

When we consider that the agents go all from the same initial node, we present in the tables 3 and 4, the results for specific cases without penalty (*SP*) and with penalty (*AP*). On the basis of the results for different groups of agents sensors deployed on the environments Map A , Map B, Island, Circle, Grid and Corridor, there is an improvement in the average number of meetings in the case with penalty compared to without penalty . It should be noted however that the approach with penalty allows to improve the overall quality strategies.

Implementation with a dispersion of agents across the graph

The tables 5 and 6 present the results where we suppose a dispersion of agents.

Table 5. Case with dispersion: meetings with five sensors agents

Name	AvMeet(SP)	#NbSol(SP)	AvMeet(AP)	#NbSol(SP)
Map A	10.0	5	13.36	11
Map B	11.93	16	13.12	33
Island	14.08	12	19.0	13
Circle	14.93	16	42.56	16
Corridor	94.13	15	110.66	15
Grid	16.07	13	19.0	24

Compared to the results in the tables 3 and 4 relative to the case without dispersion for different environments, the number of potential solutions is greater than the number of solutions of the case with dispersion (see tables 5 and 6 in both cases *SP* and *AP*). It may be due to different initial start nodes of the sensors agents. On the basis of the results for different groups of agents sensors deployed on different topologies of environments, there is an improvement in the average number of meetings in the case with penalty compared to without penalty.

Multi-objective evaluation of the multi-sensor patrol

Among the six environments that we manipulated, we have chosen to focus on Map A and Map B because we believe they are fairly

Table 6. Case with dispersion: meetings with ten sensors agents

Name	AvMeet(SP)	#NbSol(SP)	AvMeet(AP)	#NbSol(SP)
Map A	2.0	3	2.38	13
Map B	5.9	10	9.88	9
Island	2.33	3	3.33	12
Circle	5.66	21	8.96	26
Corridor	17.81	11	34.07	14
Grid	4.2	5	7.15	19

representative in terms of difficulty. We are interested in the evaluation of objective functions, as well as an estimate of the delay between the occurrence and detection of an event based on the number of ants per colony. We performed simulations on Map A and Map B environments with a variable number of agents ranging from 1*S* to 15*S*. One of the main objectives is to study the influence of the number of agents in the determination of compromise solutions. We focus here on a strategy with penalty in Map A and Map B environmentst. The figures 3 and 4 present the solutions obtained in the simulations.

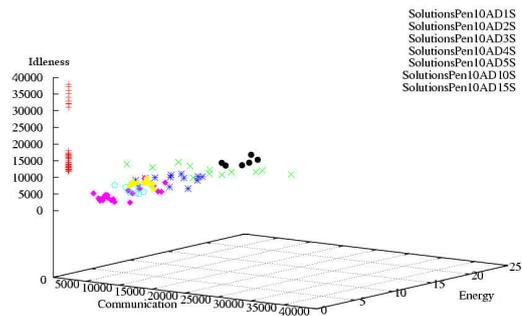


Fig. 3. Representation for Map A environment

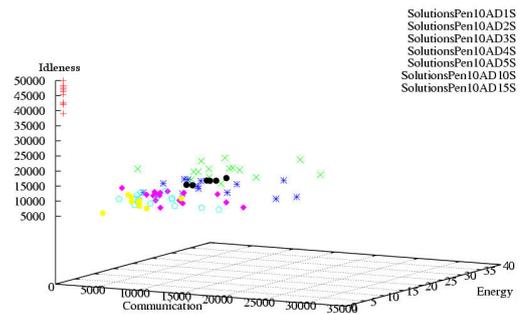


Fig. 4. Representation for Map B environment

In figure 4, the solutions for the cases four and five sensors are comparable in terms of environmental idleness and energy consumption. We observe that the more the number of sensors in the network increases, the more the communicational idleness decreases.

For a given number of sensors, the best solution will depend on the preference levels in the objective functions. Overall, we observe that the case of a network of ten sensors minimizes more communicational idleness.

The minimization of Worst Idleness, Energy consumption and Communicational Idleness are not compatible because $WI_T^\pi, CI_T^\pi,$

E_T^π can not be minimized together. It is necessary to seek compromise solutions. The set of compromise solutions is called Pareto Front.

Based on the levels of preference defined by $\text{Pref}I02T06E02: (\text{Idleness}, \text{Energy}, \text{Communication}) = (0.2, 0.2, 0.6)$, we determined compromise solutions. We chose these levels preferably with a focus on communicational idleness. But, it is possible to set other preferences with a focus on environmental idleness or energy consumed.

Figure 5 gives a graphical representation of the simulation results for the environment MapA taking into account the compromises associated with preferences defined by $\text{Pref}I02T06E02$. On this figure, $\text{PreferencePasPenAD}$, $\text{PreferencePasPenSD}$, PreferencePen10AD and PreferencePen10SD indicate compromise solutions corresponding to the preferences and associated respectively to the strategies PasPenAD (without penalty - with dispersion), PasPenSD (without penalty - without dispersion), Pen10AD (with penalty - with dispersion) and Pen10SD (with penalty - without dispersion).

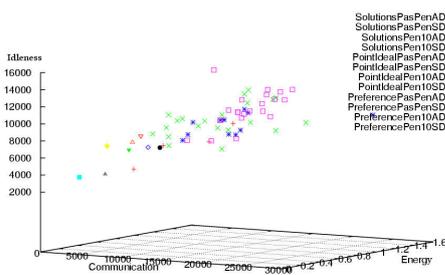


Fig. 5. Solutions for MapA with level of preference: I02T06E02

Based on these results (figure 5), we can observe that the strategy PasPeniteAD (without penalty - with dispersion) is smaller compared to the one based on communicational idleness. Contrariwise, the solutions of strategy PasPeniteAD are farther from their ideal point than the other solutions from their ideal points.

An analysis of graphic can be observed that the strategies PasPeniteAD (without penalty - with dispersion) and Pen10AD (with penalty - with dispersion) have smaller communicational idleness. The solutions of the strategies PasPeniteSD , PasPeniteAD and Pen10AD are relatively close to their respective ideal points.

6. CONCLUSION

The goal was to detect events using an ACO approach and multi-agents patrolling. The patrolling task is provided by a mobile sensors network.

The application of Ant Colony Optimization (ACO) to the multi-agent patrolling problem has been described, where agents are the mobile sensors. The ACO algorithm employs competitive colonies of ants: each colony tries to find out the best multi-agent patrolling strategy, and each ant of a colony coordinates its action with the other ants of the same colony to elaborate an individual agent's patrolling tour as short as possible. Detecting and reacting as fast as possible to some phenomena by a group of mobile sensors require that they are able to patrol while minimizing other criteria, namely the maximum communication delay between the sink and its base station and the total energy consumption over the network of mobile sensors. An evaluation of the distances between potential solution and P^* in terms of preferences has been done. For each

preference, the best solution is one that minimizes the distance to the P^* .

7. REFERENCES

- [1] M. Basseur, E. Talbi, A. Nebro, and E. Alba. Metaheuristics for Multiobjective Combinatorial Optimization Problems: Review and recent issues. In *INRIA Report-ISSN 0249-6399*, 2006.
- [2] Christina Busing, Kai-Simon Goetzmann, and Jannik Matuschke. Compromise solutions in multicriteria combinatorial optimization. In *Matheon*, 2012.
- [3] Y. Chevaleyre. Theoretical Analysis of the Multi-Agent Patrolling Problem. In , pages 302–308, 2004.
- [4] H. N. Chu, A. Glad, O. Simonin, F. Semp, A. Drogoul, and F. Charpillet. Swarm Approaches for the Patrolling Problem, Information Propagation vs. Pheromone Evaporation. In *19th IEEE International Conference on Tools with Artificial Intelligence, ICTAI, vol. 1*, pages 442–449, 2007.
- [5] Y. Colette and P. Siarry. Optimisation multiobjectif. Technical report, Edition Eyrolles, 2002.
- [6] E. T. Fute and E. Tonye. Modelling and Self-organizing in Mobile Wireless Sensor Networks: Application to Fire Detection. In *International Journal of Applied Information Systems, IJAIS, New York, USA, Vol.5 N3*, 2013.
- [7] E. T. Fute, E. Tonye, A. Koukam, and F. Lauri. Multi-Agent Patrolling Strategy: Application to the Exploration Environment Problem. In *International Journal of Automation, Robotics and Autonomous Systems, ICGST-ARAS, ISSN 1687-4811 (Print), 1687-482X (Online), Vol.9, Issue 2*, 2009.
- [8] Caroline Gagne, Marc Gravel, and Wilson L. Price. Optimisation Multi-Objectifs a l'aide d'un Algorithme de Colonie de Fourmis. In *INFOR, vol.42-1*, pages 23–42, 2004.
- [9] Arnaud Glad. *Etude de l'auto-organisation dans les algorithmes de patrouille multi-agent fondes sur les phromones digitales*. PhD thesis, Universit Nancy 2, 2011.
- [10] Tristram Grabener and A. Berro. Optimisation multiobjectif discrete par propagation de contraintes. In *Actes JFPC*, 2008.
- [11] F. Lauri and A. Koukam. A Two-Step Evolutionary and ACO Approach for Solving the Multi-Agent Patrolling Problem. In , *Hong-Kong, China*, 2008.
- [12] A. Machado, G. Ramalho, and . Multi-Agent Patrolling : an Empirical Analysis of Alternatives Architectures. In *3rd* , pages 155–170, 2002.
- [13] J. B. Seung, Gustavo de Veciana, and Xun Su. Minimizing Energy Consumption In Large-scale Sensor Networks Through Distributed Data Compression And Hierarchical Aggregation. In *IEEE Journal on Selected Areas in Communications*, pages 1–10, 2004.
- [14] Elie Fute T., E. Tonye, A. Koukam, and F. Lauri. Strategie de Patrouille Multi-Agents : Application au Probleme d'Exploration d'un Environnement. In *5th IEEE International Conference: Sciences of Electronic, Technologies of Information and Telecommunications*, 2009.
- [15] Jurgen Teich. Pareto-Front Exploration with Uncertain Objectives. In *Springer-Verlag, Berlin Heidelberg*, pages 314–328, 2001.