A game theory-based approach for robots deploying wireless sensor nodes

Ines Khoufi, Pascale Minet, Mohamed-Amine Koulali, Mohammed Erradi

A Game Theory-Based Approach for Robots Deploying Wireless Sensor Nodes

Ines Khoufi*, Pascale Minet*, Mohamed-Amine Koulali† and Mohammed Erradi‡
*INRIA, Rocquencourt, 78153 Le Chesnay Cedex, France.
†Ecole Nationale des Sciences Appliquees, Université Mohammed Premier, BP 669 Al Qods, Oujda, Morocco.
‡MIS Lab., ENSIAS, Mohammed V-Souissi University, Madinat Al Irfane BP 713 Agdal, Rabat, Morocco.
ines.khoufi@inria.fr, pascale.minet@inria.fr, ma.koulali@ump.ma and erradi@ensias.ma

***Abstract*—Wireless Sensor Networks (WSNs) are deployed in many fields of application. Depending on the application requirements, sensor nodes can either be mobile and autonomous or static. In both cases, they are able to cooperate together in order to monitor a given area or some given Points of Interest (PoIs). Static sensor nodes need one or several agent(s) (humans or robots) to deploy them. In this paper, we focus on the deployment of static sensor nodes in an area containing obstacles, using two mobile robots. We want to minimize the time needed by the two robots to deploy all the sensor nodes and to return to their starting position. We require that each sensor node is placed at a PoI position, no PoI position is empty and no PoI position is occupied by more than one sensor node. The problem consists in determining the best strategy for each robot in order to meet these constraints. We adopt a game theory approach to solve this problem.**

***Index Terms*—WSN, game theory, coverage, connectivity, points of interest, relay nodes, assisted deployment, robot tour, obstacles.**

I. INTRODUCTION AND MOTIVATION

Many monitoring applications rely on Wireless Sensor Networks, WSNs, in various fields of application (e.g. predictive maintenance, industrial processes, environment, e-health, precision agriculture) to gather data on the environment monitored. Such WSNs need to satisfy coverage and connectivity requirements to ensure data gathering with a good quality of service. In this paper, we focus on the coverage of Points of Interest, (PoIs). These PoIs are sources of information that is of major importance for the monitoring process. As a consequence, a sensor node must be placed at the position of each PoI. However, this in itself is not sufficient, as the collected information must reach the sink to be analyzed. That is why connectivity of each PoI with the sink is necessary and to meet this requirement, additional relay nodes must be placed at appropriate locations.

The advantage of WSNs compared to wired networks is their ease of deployment. We can distinguish two types of WSNs depending on the type of wireless nodes:

- Those based on autonomous and mobile wireless nodes: in such a case a self-deployment algorithm is adopted where wireless nodes move to the positions that are computed. This computation, usually iterative, can be centralized to avoid expensive node oscillations. In a second step, wireless nodes move directly to their final positions. This computation can also be distributed to take advantage of the processing capability of each node to dynamically adapt to the real environment and wireless nodes move at each iteration.
- Those based on static wireless nodes: in this case, mobile agents (robots or humans) are needed to place the wireless nodes at their final positions, which have been computed previously. This is termed an assisted deployment.

In this paper, we focus on the deployment of wireless nodes assisted by robots. The deployment duration should be minimized in order to save the robot battery and in case of hostile environment, such as a post-disaster situation (e.g. fire, pollutant leak, radiation), to prolong its lifetime. Data sent from the PoIs and collected at the sink will allow us to assess damage, and the time needed for this damage assessment must be minimized. We again distinguish two cases:

- A single robot is used. The problem in this case is an optimization problem where the optimized tour of the robot has to be found. This is a variant of the Traveling Salesman Problem (TSP), where the changes of direction of the robot have to be taken into account in the tour duration as computed in Eq. 1 (see [1] for examples).
- Several robots are available. The existence of several robots allow them to work in parallel. This reduces the workload assigned to each of them and hence minimizes the deployment duration as well as the time needed for damage assessment.

In this paper we focus on two concurrent robots. The problem is considered as a game theory problem with the robots being two non-cooperative players. The goal is to identify the winning strategies for each robot. A strategy is the ordered set of PoIs visited by a robot. The payoff of each strategy must be computed in order to favor the coverage of each PoI by a single robot. Consequently, a robot will be penalized when either a PoI is not visited by a robot or when it is visited by both robots. In this non-cooperative game, each robot tends to choose the strategy maximizing its payoff.

To be more representative of a real environment, we have to take into account the existence of obstacles. An obstacle may block the robot path, and may prohibit (depending on its type) communication between neighboring nodes. A transparent obstacle has no impact on wireless communication, however an opaque obstacle prohibits wireless communication between neighboring nodes which are not in the line of sight of each other making opaque obstacles more challenging than transparent ones. Thus, we have to deal with two problems caused by obstacles:

- Bypassing obstacles: a strategy is proposed in Section III-C to allow the robot to move from one PoI position to another while avoiding one or several obstacles.
- Maintaining network connectivity: Opaque obstacles should also be taken into account when computing the positions of relay nodes. Two consecutive relay nodes may be at a distance less than or equal to their communication range, yet unable to communicate due to the presence of an opaque obstacle between them. In this case, additional relay nodes may be needed to ensure network connectivity.

II. State of the art

In WSNs, a mobile robot can be used either to deploy/redeploy static sensor nodes or to collect data from sensor nodes already deployed. In [2], the authors propose a solution using a mobile robot to ensure coverage of an area containing obstacles. This mobile robot is in charge of deploying the minimum number of sensor nodes along its trajectory such that neighboring nodes are at an optimal distance. The robot is assumed to know the dimensions of the area as well as the shapes obstacles and it deploys sensor nodes while exploring the whole area. Another problem in WSNs is to deploy sensor nodes in specific zones or points of interest. The PoI positions are predetermined, and the robot has to deploy a sensor node at each PoI. However, the order in which the robot visits the PoIs may have an impact on the distance covered or the duration of robot tour. In this context, the authors in [3] refer to the traveling salesman problem with neighborhoods to find the minimum path in terms of distance traveled by the robot. The robot should visit each zone of interest only once and return to its starting position. To solve this problem, they propose an iterative algorithm where a non visited zone of interest is added at each iteration such that the distance added to robot tour is minimum. The authors propose the same algorithm for the robot to stop at some pause points in order to collect data from sensor nodes. thus, the robot tour is an ordered list of pause points. This tour should, on the one hand, allow the robot to communicate with every sensor node during the tour with a minimum number of pause points, and on the other hand, it should have the minimum length in terms of distance traveled.

In a previous study [1], we proposed a Robot Deploying Sensor nodes problem (RDS) which aims to minimize the robot tour duration (i.e. the time spent by the robot to place its sensor nodes and return to its starting position). Unlike the TSP problem, which minimizes the distance traveled by the robot, our RDS problem takes into account the time the robot spends in changing direction. We solved this problem using optimization algorithms: a genetic algorithm and a 2-opt algorithm. We proposed a hybrid algorithm that combines both genetic and 2-opt algorithms and provides better results in terms of tour duration.

In this work we consider a game theory approach for robot assisted sensors deployment. Non cooperative game theory provides a mathematical framework for modeling, analyzing and predicting the outcome of strategic interactions involving interdependent rational players [4]. In strategic decision taking, conflicts of interests arise and own decisions are no longer sufficient to ensure optimal payoff. Indeed, the outcome of the game will depend on the whole strategy profile (set of actions) of all the actors. Game theory has strong roots in economics, but has also been widely applied in several other fields such as political science [5], biology [6], engineering and computer science [7]. The past decade witnessed a surge in research effort to apply game theory in the context of modern communication systems [8], [9] and [10]. Indeed, there has been an exponential growth of connected devices combined with an increase of their communication and computational capabilities. Game theory provides a formal framework for distributed decision while reducing the overhead of protocol design inherent to central authorities. Games in strategic form are well suited for the study of N player games with discrete action space. An important solution concept for such games is the Nash Equilibrium [11]. A Nash equilibrium is a strategy profile to which all players have to commit. For instance, none of the players will have incentive to deviate from the Nash equilibrium to increase its payoff. The authors of [12] proposed a game-theoretic formulation for activity scheduling of sensors. The problem is to arrange the sensing schedule of sensors to maximize average area coverage. The considered game is proven to be a potential/congestion game with a pure Nash equilibrium. The problem of minimizing the probability of undetected intrusion, in environments with obstacles is considered in [13]. The problem is equivalent to partial barrier coverage formulated as a zero-sum game between the intruder and the deployment of the guards. The saddle point of the game (Equilibrium) is proven to exist for mixed strategies only.

In this paper, we focus on the deployment of static sensor nodes at specific positions (PoIs) using two robots that proceed in parallel in order to minimize the deployment duration. We require that each sensor node is placed at a PoI position, no PoI position is empty and no PoI position is occupied by more than one sensor node. The problem consists in determining the best strategy for each robot in order to meet these constraints. We adopt a game theory approach to solve this problem and find the best strategy for each robot. We consider a realistic area with some obstacles and show how to cope with these obstacles.

III. Problem definition

A. Goal

Our goal is to minimize the tours duration of two mobile robots in charge of deploying static sensor nodes in an environment with known obstacles. Each sensor node must be positioned at a Point of Interest (PoI). To maintain network connectivity, the robot will also deploy additional relay nodes during its tour, Then, each PoI will be able to communicate with the sink. The problem consists in determining the tours of the two robots that minimize the deployment duration while ensuring the coverage of each PoI by exactly one robot.

Figure 1 depicts the tour of two robots where each PoI is visited by exactly one robot and relay nodes are placed to ensure connectivity of each PoI with the sink.

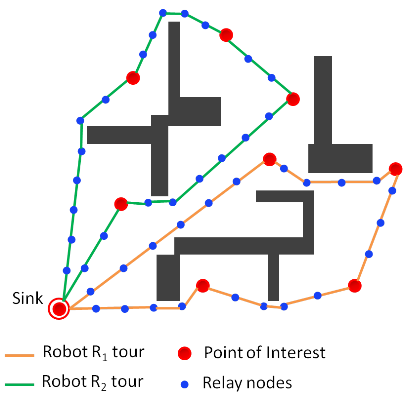


Fig. 1: Robot tours and relay node deployment in the presence of obstacles

B. Assumptions and definitions

- We assume that each robot R_i, $i = 1$ or 2, knows:
  - n, the number of PoIs. Each PoI is denoted P_i, for $i \in [1, n]$.
  - The position of each PoI.
  - The number, position and shape of the obstacles.
  - The area considered.
- Each mobile robot is able to know its position and to move to a given position.
- Both robots have the same linear speed ls and the same angular speed as. As an example, in the simulation reported in this paper, we take $ls = 1m/s$ and $as = 10/s$.
- Each robot R_i has the capacity to carry $C_{max,i}$ sensor nodes.
- Each robot R_i has the capacity to carry $C_{relaymax,i}$ relay nodes.
- Both robots have the same starting position, the sink denoted P_0 for simplicity, and should return to this position.
- Let S_i denote the set of strategies played by robot R_i. Any strategy s_i played by R_i is defined by the ordered set of PoIs visited by R_i.
- To cope with obstacles, a bypassing approach is adopted as explained in the nest section.

C. Deployment duration and obstacles

The deployment duration D_i of robot R_i depends not only on the time needed to travel a distance but also on the time needed to carry out changes in direction. Hence, we compute D_i for any strategy s_i as follows:

$$D_i = \sum_{j \in s_i} d_{j,j+1}/ls + \sum_{j \in s_i} a_{j-1,j,j+1}/as. \tag{1}$$

Where j and $j+1$ are two successive PoIs in s_i. $d_{j,j+1}$ is the distance between two successive PoIs in s_i. $a_{j-1,j,j+1}$ is the angle formed by the segments $[j-1, j]$ and $[j, j+1]$ corresponding to three successive PoIs in s_i. We notice however that the tour duration is the same when the robot visits the same nodes but in the reverse order.

If there exists an obstacle between the sensor node positions A and B, the direct path from A to B is made impossible as illustrated in Figure 2. We propose to define intermediate positions I_i that allow the robot to reach B avoiding the obstacle. We replace the impossible direct path between A and B by a possible one that can be seen as a juxtaposition of segments $I_i I_{i+1}$. The tour duration due to this path is computed as the sum of the duration due to any segment composing the path. Figure 2 shows the two intermediate points I_1 and I_2 that are introduced in the path taken by the robot to reach B from A. The path A, I_1, I_2, B replaces the direct path A, B.

Additional intermediate positions are introduced in the tour of the robot to bypass obstacles. The deployment duration takes into account these intermediate positions. Furthermore, additional relay nodes are needed to ensure connectivity in case of opaque obstacles. The positions of these relay nodes are also taken into account.

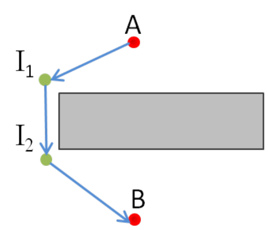


Fig. 2: Intermediate points between sensor node positions A and B

IV. Problem Formalization

We proceed in two steps. In the first step, we focus on the coverage problem: the goal is only to ensure the coverage of all PoIs while minimizing the tour duration of each robot. In other words, only one robot must place a sensor node at each PoI. In the second step, we are interested in both coverage and connectivity: each robot must, in addition, place relay nodes to ensure connectivity of each PoI with the sink.

A. Coverage problem

We can now model our coverage problem as a non cooperative game. A game is defined by the set of players, the set of strategies for each player and the set of payoffs corresponding to the strategies used by each player. In our case, the two players are the mobile robots. The action of each robot consists in selecting the next PoI to visit in its tour. Consequently the strategy s of a robot is an ordered set of PoIs visited by this robot.

Let R_1 and R_2 denote our two robots. Let $\{P_1, P_2, .., P_n\}$ denote the set of n PoIs that should be monitored. Let $\mathcal{P}_i(s_i, s_{-i})$ denote the payoff of player R_i when it plays strategy s_i while the other player plays strategy s_{-i}. The payoff computation follows some rules:

- To incite robots to visit all PoIs exactly once, the payoff of a robot increases when the number of PoIs visited increases. However, the payoff of both robots becomes negative if some PoIs are visited by either no robot or both robots.
- To minimize the tours duration of robots, the payoff of a robot increases for a given set of PoIs visited when the tour duration of the robot decreases.

Algorithm 1 shows how to calculate the payoff of one robot in charge of ensuring PoIs coverage.

We use a weight factor α higher than D_i to model positive outcome values.

Algorithm 1 Calculate $\mathcal{P}_i(s_i, s_{-i})$ for Coverage problem

```
if (Number of PoIs visited by both R_i and R_{-i} <> 0) then
    P_i(s_i, s_{-i}) = -1
else
    if ((C_{max,i} + C_{max,-i}) >= n) then
        if (Number of PoIs visited by neither R_i nor R_{-i} <> 0)
        then
            P_i(s_i, s_{-i}) = -1
        else
            P_i(s_i, s_{-i}) = α / D_i
        end if
    else
        if (Number of PoIs visited by R_i < C_{max,i}) then
            P_i(s_i, s_{-i}) = -1
        else
            P_i(s_i, s_{-i}) = α / D_i
        end if
    end if
end if
```

Consequently each player R_i wants to maximize its payoff $\mathcal{P}_i(s_i, s_{-i})$. Under the constraints:

- $C_i \leq C_{max,i}$, where C_i is the number of sensor nodes carried by the robot R_i.

B. Coverage and connectivity problem

In the coverage and connectivity problem, each robot places a relay node each time it travels a distance $Dist$. The coverage and connectivity problem differs from the coverage problem by an additional constraint on $C_{relay,i}$ the number of relay nodes placed by a robot R_i. We must have:

- $C_{relay,i} \leq C_{relaymax,i}$.

Strategies violating this constraint are eliminated.

The payoff of any strategy is computed as given in Algorithm 1.

V. Problem resolution

Notice that in both games, the payoff computed for player R_i depends not only on s_i the strategy chosen by R_i but also on the strategy s_{-i} chosen by the other player R_{-i}.

A strategy profile (s_i^*, s_{-i}^*) is a Nash equilibrium if and only if no unilateral deviation of the strategy of a single player is profitable for that player. Hence, $\forall i, \forall s_i \in S_i, \mathcal{P}_i(s_i^*, s_{-i}^*) \geq \mathcal{P}_i(s_i, s_{-i}^*)$.

Nash proved the existence of at least one Nash equilibrium when mixed strategies are allowed in a game with a finite number of players and each player chooses among pure strategies.

Both problems are solved in a similar way:

- Determining all strategies for each player.
- Eliminating all strategies violating the constraints. The remaining strategies are the valid strategies of any player.
- Computing the payoff for all the possible combinations of valid strategies for all players.
- Computing the Nash equilibrium using the Gambit tool [14].

The number of strategies for robot R_i visiting q PoIs with $q \leq min(C_{max,i}, n)$ is: $C_n^q * \frac{q!}{2}$. This is because the strategies $\{P_j, P_{j+1}...P_{j+m}\}$ and $\{P_{j+m}..., P_{j+1}, P_j\}$ have the same payoff.

Hence the total number of valid strategies for robot R_i is equal to:

$$\sum_{q \in \{1, C_{max,i}\}} C_n^q * \frac{q!}{2}.$$

A. Coverage problem

For any given strategy s_{-i}, the strategy s_i that maximizes $\mathcal{P}_i(s_i, s_{-i})$ in the coverage problem consists of visiting only all the PoIs that are not visited by R_{-i}, provided it meets the constraint $C_{max,i}$ and selecting the visit order that minimizes the deployment duration Di.

We first notice that if $C_{max,i} < \lceil n/2 \rceil$ for any $i \in [1, 2]$, it is impossible to cover all the PoIs with two robots.

In any other case, we obtain a Nash equilibrium where each PoI is visited exactly once, ensuring the full coverage of all the PoIs without any redundancy.

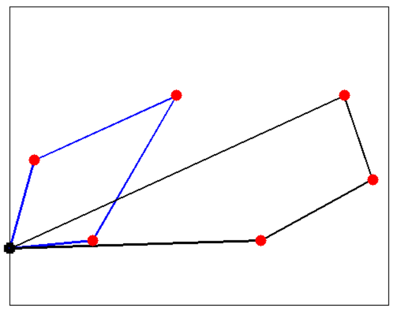
In this section, we evaluate the tour duration of the two robots deploying sensor nodes in an area with and without obstacles. We start by computing the tour duration for various values of $C_{max,i}$ and $C_{max,-i}$. The sum of $C_{max,i}$ and $C_{max,-i}$ should be higher than or equal to the number of PoIs to be covered. Then, we evaluate the duration of both tours in different configurations. These configurations are different in terms of the number and shape of the obstacles in the area considered.

1) Area without obstacles: To evaluate the impact of the robot capacity to carry sensor nodes, we vary the values of $C_{max,i}$ and $C_{max,-i}$.

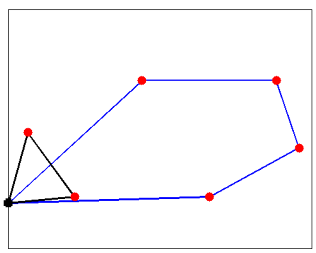
When $C_{max,1}$ is equal to $C_{max,2}$ and their sum is equal to the number of PoIs (See Case 1 in Table I and Figure 3a))

	Case 1 fig. 3a)		Case 2 fig. 3b)		Case 3 fig. 3c)	
	Robot 1	Robot 2	Robot 1	Robot 2	Robot 1	Robot 2
$C_{max,i}$	3	3	4	2	4	4
E. duration	644.54	1056	1090.4	366.23	1090.4	366.23
PoIs visited	3	3	4	2	4	2

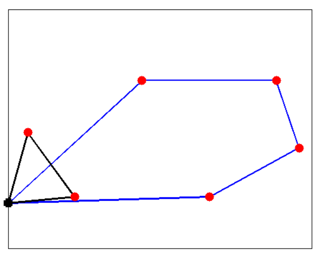
TABLE I: Impact of $C_{max,i}$ on the deployment duration.



a) $C_{max,i} = 3$, $C_{max,-i} = 3$.



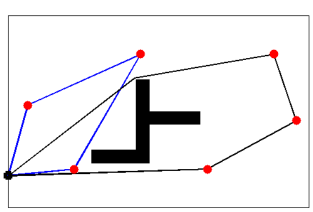
b) $C_{max,i} = 4$, $C_{max,-i} = 2$.



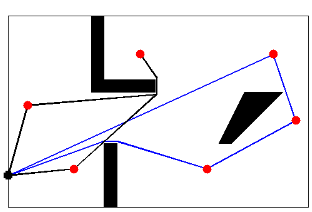
c) $C_{max,i} = 4$, $C_{max,-i} = 4$.

Fig. 3: Best robot tours with different $C_{max,i}$ (No obstacles).

the Nash equilibrium provides two robot tours that visit all PoIs exactly once. It may not be possible to find two robot tours with the same duration due to the position of the PoIs in the area considered. However, the ones computed are the best combination to minimize the tour duration of robots.
In Figure 3b) and Figure 3c), we get the same Nash equilibrium, where the first robot visits $C_{max,i}$ PoIs and the second one visits the PoIs not visited by the first one.



a) $C_{max,i} = 3$, $C_{max,-i} = 3$.



b) $C_{max,i} = 3$, $C_{max,-i} = 3$.

Fig. 4: Impact of obstacles on the best robot tours.

2) Area with obstacles: In Figure 4a) and Figure 4b), we observe that the presence of obstacles modifies the trajectory of the robots. We obtain a Nash equilibrium for the two obstacle configurations, where each robot visits 50% of PoIs. To bypass the obstacle, each robot uses intermediate points. For this reason, we notice that the tour duration is increased: 1071 in Configuration 1 Figure 4a) and 1077 in Configuration 2 Figure 4b) instead of 1056 as shown in Table II.

	Configuration 1 fig. 4a)		Configuration 2 fig. 4b)	
	Robot 1	Robot 2	Robot 1	Robot 2
$C_{max,i}$	3	3	3	3
Deployment duration (s)	644.54	1071.3	1077.1	786.69
PoIs visited	3	3	3	3

TABLE II: Impact of the presence of obstacles on the deployment duration.

B. Coverage and connectivity problem

For any given strategy s_{-i}, the strategy s_i that maximizes $\mathcal{P}_i(s_i, s_{-i})$ in the coverage and connectivity problem consists in visiting only all the PoIs that are not visited by R_{-i}, provided it meets the constraints $C_{max,i}$ and $C_{maxrelay,i}$, while selecting the visit order that minimizes the Deployment duration Di.

Notice that the robot places relay nodes along its trajectory such that two node-disjoint paths are established between each PoI and the sink. Then, a robust network connectivity is maintained.

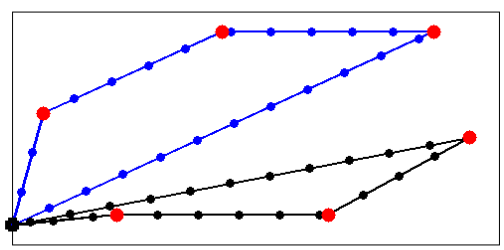


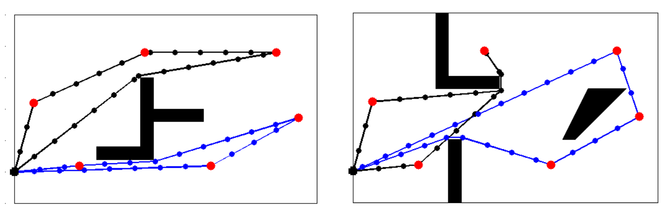
Fig. 5: $C_{max,i} = 3$, $C_{max,-i} = 3$

1) Area without obstacles: In this section, we assume that the robot is able to carry a limited number of relay nodes $C_{maxrelay,i}$. In Figure 5, $C_{maxrelay,i} = 22$. The two robot tours are different from the ones illustrated in Figure 4 where there is no constraint on the number of relay nodes. The two robot tours in Figure 4 provide a smaller deployment duration than those of Figure 5. However, they require a number of relay nodes higher than $C_{maxrelay,i}$. Then, the result obtained in Figure 5 presents the better combination taking into account both the number of relay nodes and the deployment duration.

	Without Obstacles fig. 5	
	Robot 1	Robot 2
$C_{max,i}$	3	3
Deployment duration (s)	995.6	947.6
PoIs visited	3	3
$C_{maxrelay,i}$	22	22
$C_{relay,i}$	22	21

TABLE III: The deployment duration with relay nodes.

2) Area with obstacles: When obstacles exist the length of each tour may increase due to obstacle bypassing. Then, the value of $C_{maxrelay,i}$ may increase according to the obstacle configuration. We fix $C_{maxrelay,i} = 23$ for Configuration 1 and $C_{maxrelay,i} = 24$ for Configuration 2, respectively (see Table IV). We observe in Figure 6a) that the two robot tours are different from those in Figure 4a) due to the constraint on $C_{maxrelay,i}$. However, in Configuration 2 (see Figures 4b) and 6b)), the two robot tours do not change, whether we take into account the relay nodes or not. When we try to decrease $C_{maxrelay,i}$ to a value less than 24, we do not get any Nash equilibrium as all strategies are eliminated due to a violation of the $C_{maxrelay,i}$ constraints.



a) $C_{max,i} = 3$, $C_{max,-i} = 3$. b) $C_{max,i} = 3$, $C_{max,-i} = 3$.

Fig. 6: Impact of obstacles on the best robot tours with relay nodes placement.

	Configuration 1 fig. 6a)		Configuration 2 fig. 6b)	
	Robot 1	Robot 2	Robot 1	Robot 2
$C_{max,i}$	3	3	3	3
Deployment duration (s)	951.5	1014	1077.1	786.69
$C_{maxrelay,i}$	23	23	24	24
$C_{relay,i}$	21	23	24	19

TABLE IV: The Deployment duration with relay nodes and obstacles.

VI. Conclusion

In this paper we focused on the deployment of wireless sensor nodes by mobile robots. The goal is to ensure the coverage of PoIs and maintain connectivity of each PoI with the sink. To be closer to real conditions, we take into account the presence of obstacles. We want to find the trajectories of two mobile robots which satisfy the following constraints: all the PoIs are visited exactly once and the tour duration of the robots is minimized. Game theory is used to formalize this problem as a non-cooperative game with two players, to find the Nash equilibria for various configurations. We studied the impact of obstacles on the deployment duration. The robot tours may differ depending on whether obstacles are present or not. However, the Nash equilibrium obtained, provides the best combination of the two robot tours.

Acknowledgment

This work has been partially funded by the Cluster Connexion project. For more details see http://www.cluster-connexion.fr/.

Special thanks to the members of the AWSN project for their cooperation, see http://hipercom.inria.fr/euromed/.

References

[1] I. Khoufi, E. Livolant, P. Minet, M. Hadded, A. Laouiti, *Optimized trajectory of a robot deploying wireless sensor nodes*, Wireless Days 2014, Rio de Janeiro, Brazil, November 2014.

[2] C. Y. Chang, H. R. Chang, C. C. Hsieh and C. T. Chang*OFRD:Obstacle-Free Robot Deployment Algorithms for Wireless Sensor Networks*, WCNC 2007, Kowloon, March 2007.

[3] Y. Wang and C. Wu, *Robot-Assisted Sensor Network Deployment and Data Collection*, IEEE International Symposium on Computational Intelligence in Robotics and Automation, Jacksonville, FL, USA, June 20-23, 2007.

[4] T. Basar, G. J. Olsder, G. Clsder, T. Basar, T. Baser, and G. J. Olsder, *Dynamic non-cooperative game theory*, vol. 200, 1995.

[5] J. D. Morrow, *Game theory for political scientists*, Princeton University Press Princeton, NJ, 1994.

[6] P. Hammerstein and R. Selten, *Game theory and evolutionary biology*, Handbook of game theory with economic applications, vol. 2, pp. 929-993, 1994.

[7] K. R. Apt and E. Gradel, *Lectures in game theory for computer scientists*, 2011.

[8] S. Haykin, *Cognitive radio: brain-empowered wireless communications*, IEEE Journal on Selected Areas in Communications, vol. 23, no. 2, pp. 201-220, 2005.

[9] S. Koskie and Z. Gajic, *A nash game algorithm for sir-based power control in 3g wireless cdma networks*, IEEE/ACM Transactions on Networking (TON), vol. 13, no. 5, pp. 1017-1026, 2005.

[10] T. Alpcan, T. Basar, and S. Dey, *A power control game based on outage probabilities for multicell wireless data networks*, IEEE Transactions on Wireless Communications, vol. 5, no. 4, pp. 890-899, 2006.

[11] J. F. Nash et al., *Equilibrium points in n-person games*, Proceedings of the national academy of sciences, vol. 36, no. 1, pp. 48-49, 1950.

[12] X. Ai, V. Srinivasan, and C.-K. Tham, *Optimality and complexity of pure nash equilibria in the coverage game*, IEEE Journal on Selected Areas in Communications, vol. 26, no. 7, pp. 1170-1182, 2008.

[13] S. Kloder and S. Hutchinson, *Partial barrier coverage: Using game theory to optimize probability of undetected intrusion in polygonal environments*, in IEEE International Conference on Robotics and Automation, 2008, pp. 2671-2676.

[14] Gambit: a software tool for game theory, http://gambit.sourceforge.net/