

Path Planning of Mobile Sinks in Charge of Data Gathering: a Coalitional Game Theory Approach

Ines Khoufi, Pascale Minet, Mohamed-Amine Koulali, Abdellatif Kobbane

► **To cite this version:**

Ines Khoufi, Pascale Minet, Mohamed-Amine Koulali, Abdellatif Kobbane. Path Planning of Mobile Sinks in Charge of Data Gathering: a Coalitional Game Theory Approach. 35th IEEE – International Performance Computing and Communications Conference, Dec 2016, Las Vegas, United States. hal-01410082

HAL Id: hal-01410082

<https://hal.inria.fr/hal-01410082>

Submitted on 6 Dec 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Path Planning of Mobile Sinks in Charge of Data Gathering: a Coalitional Game Theory Approach

Ines Khoufi, Pascale Minet
Inria, 2 rue Simone Iff, CS 42112,
75589 Paris Cedex 12, France
Email: ines.khoufi@inria.fr
pascale.minet@inria.fr

Mohamed-Amine Koulali
MSN Research Team, ENSAO
Mohammed I University Oujda, Morocco
Email: m.koulali@ump.ac.ma

Abdellatif Kobbane
SIME Lab. ENSIAS
Mohammed V University Rabat, Morocco
Email: kobbane@ensias.ma

Abstract—Game theory is often used to find equilibria where no player can unilaterally increase its own payoff by changing its strategy without changing the strategies of other players. In this paper, we propose to use coalition formation to compute the optimized tours of mobile sinks in charge of collecting data from static wireless sensor nodes. Mobile sinks constitute a very attractive solution for wireless sensor networks, WSNs, where the application requirements in terms of node autonomy are very strong unlike the requirement in terms of latency. Mobile sinks allow wireless sensor nodes to save energy. The associated coalition formation problem has a stable solution given by the final partition obtained. However, the order in which the players play has a major impact on the final result. We determine the best order to minimize the number of mobile sinks needed. We evaluate the complexity of this coalitional game as well as the impact of the number of collect points per surface unit on the number of mobile sinks needed and on the maximum tour duration of these mobile sinks. In addition, we show how to extend the coalitional game to support different latencies for different types of data. Finally, we formalize our problem as an optimization problem and we perform a comparative evaluation.

I. CONTEXT AND MOTIVATIONS

In Wireless Sensor Networks (WSNs) where sensor nodes are static, we distinguish three architectures for data gathering. In the first architecture, the sink is static. Collected data are transmitted in multi-hop to the sink using a routing tree. The drawback of this architecture is the heavy load of nodes close to the sink, leading to a high energy consumption for these nodes. In the second architecture, there is one or several mobile sinks. Each sensor node directly transmits its data to the mobile sink that visits it. The frequency of these visits must be tuned to meet the latency requirement of the application considered. For such applications that do not have strong latency requirements, intermittent connectivity is sufficient. The problem becomes, how to plan the optimized path of these mobile sinks in order to meet the latency constraint. The third architecture is an hybrid of the previous ones. In this paper, we focus on the second architecture.

The concept of mobile sink has been introduced to avoid that nodes close to the static sink receiving a lot of traffic quickly exhaust their battery. It leads to a more uniform energy consumption among nodes. From the communication point of view, each wireless sensor node is only in charge of transmitting its own data to the mobile sink when it is visiting

it. Compared to a network with permanent connectivity, each node spares the energy that it would use in forwarding data received from its neighbors. Hence, a mobile sink by saving energy of nodes maximizes network lifetime. This is crucial for WSNs that are deployed in areas that are either hostile (e.g. radioactivity, corrosive atmosphere) or whose access is difficult, dangerous or expensive. It is then very important to optimize the tour duration of the mobile sink in charge of visiting the wireless sensor nodes. Furthermore, since the worst latency delay is obtained when data are ready to be transmitted on the wireless sensor node just when the mobile sink left this node, data have to wait a tour duration before being transmitted to the mobile sink. Subsequently, these data have to wait another tour duration before being delivered to the static sink in charge of processing the data collected. Hence, the worst latency is equal to two tour durations. An easy method to minimize the tour duration is to use several mobile sinks instead of a single one. The problem becomes how to compute the required number of mobile sinks, how to distribute the sensor nodes to visit among the mobile sinks in such a way that data delivery latency is met (i.e. data are delivered to the static sink before their deadline). The basic idea of this paper is to use game theory as a framework to state our problem and find the optimized number of mobile sinks needed as well as the optimized tour performed by each mobile sink. The remainder of this paper is organized as follows. In Section II, we briefly present studies related to mobile sinks and describe some uses of game theory in wireless networks. In section III, we model the data gathering by mobile sinks as a coalitional game with n players, where each player is a static wireless sensor node to be visited. Section IV reports our simulation results as well as complexity results of the approach proposed. We show in Section V how to extend this approach to deal with heterogeneous latency requirements or heterogeneous sampling rates for the data produced by static sensor nodes. In Section VI we compare the coalition game with an optimization heuristic. Finally, we conclude in Section VII.

II. RELATED WORK

In the literature, several studies are proposed to ensure data gathering using mobile sink. We distinguish three types of approaches. In the first one, the sink moves to the location

inside the covered region that minimizes the energy consumed by all sensor nodes during data gathering [1]. In the second approach, the sink follows a free or random trajectory. Data are stored in specific nodes to be later sent to or retrieved by a mobile sink [2]. In the third approach, the mobile sink trajectory is computed in advance. This trajectory is optimized to take into account energy and latency constraints. If only one sink is considered, this problem refers to the Traveling Salesman Problem (TSP) [3], [4]. If several sinks are used, this problem shares some similarities with the Vehicle Routing Problem (VRP) [5]. In VRP, the problem consists in determining the fleet routes to deliver goods from a depot to customers while minimizing the vehicles total travel cost. In our study, we compute the optimized number of mobile sinks and optimize their trajectories in terms of duration. Unlike other cited studies, we do not assume that the WSN is connected.

In a previous work, we used optimization techniques to solve a similar problem. In this paper, we propose a solution based on a coalition game and evaluate its performances with regard to genetic algorithm. Game theory is a mathematical discipline that investigates the strategic scenarios where rational agents are acting to maximize their payoffs. The theory covers both cooperative [6] and non-cooperative [7] settings depending on the agents willingness to cooperate. In strategic interactions, the gain of a particular player does not depend solely on its own decision. Indeed, the behavior of other involved players will highly impact the outcome of the game (i.e., individual payoffs). In [12] the mobility pattern of mobile sinks in Wireless Sensor Network for efficient gathering is proposed. Multiple sinks play a coordination game with the common objective of covering the network while minimizing the sinks moves. In a previous work [13], we considered the deployment of wireless sensor nodes using mobile and autonomous robots. We formulated this problem as a non-cooperative game between two players that are the mobile robots. Each user has different strategies, each strategy corresponding to the ordered list of the sensor nodes visited by the mobile robot considered. We found the Nash equilibrium corresponding to the optimized tours performed by each robot. However, this approach is not scalable. That is why, here we adopt a totally different approach that does not suffer from the scalability problem. When cooperation is possible, rational players form coalitions to maximize their gains. Later, the value of a coalition (achieved gain) can either be divided among its members (i.e., transferable utility cooperative games) or not (non-transferable utility cooperative games). Hedonic games are a particular class of cooperative games where decision of a player to join a given coalition depends only on its members. Structural properties of the value function such as super-additive are key parameters that govern the coalition formation processes. In hedonic games with non super-additive value function the grand coalition will seldom form and hence the partition of the players into coalitions is studied and its stability investigated. In [8], authors investigate incentive mechanisms for delay tolerant networks. Their

objective is to provide reward for relays that deliver content to destination. Relays form coalitions to reduce their energy consumption while increasing the encounter probability. The authors propose a transferable utility coalitional game to model the studied problem and study the coalitions stability in terms of the Shapley value. The behavior of mobiles and their cooperation in the transmission of contents is investigated through Imitative Boltzmann-Gibbs Learning Algorithm. In [11], a delay tolerant network where mobile nodes collect packets from data sources and deliver them to a sink is considered. The nodes are rewarded through a wireless energy transfer from the sink. The authors provide a two step formulation of the problem. First the optimal packet delivery policy of each mobile node is derived. Then, a repeated coalition formation game, is proposed to investigate the cooperation strategies of multiple mobile nodes. In [10], a coalition game model for self-organizing unmanned aerial vehicles collecting data from randomly located tasks in wireless networks was introduced. A Hedonic game formulation was provided and the stability of the coalition formation game proved.

III. PROBLEM STATEMENT

A. Notations and definitions

Definition 1: A normal form cooperative game is a couple (\mathcal{N}, v) where:

- \mathcal{N} is a set of players.
- v is a value function that assigns a real value to every coalition $C \in 2^{\mathcal{N}}$.

A coalition formation game models groups of agents or players acting together. It is defined by a set of n rational players, denoted $\{i, i \in [1, n]\}$ that form coalitions in order to increase their payoff.

For any player $i \in [1, n]$, let \mathcal{C}_i denote the set of all the coalitions containing i and let $u_i(C)$ be the payoff of player i when belonging to coalition C .

Each player i builds its **preference order**, denoted $>_i$ over its set of coalitions \mathcal{C}_i . Let C_1 and C_2 denote two coalitions in \mathcal{C}_i , C_2 is strictly preferred to C_1 by i , denoted $C_2 >_i C_1$, if and only if $u_i(C_2) > u_i(C_1)$.

According to [10], a coalition formation is said **hedonic**, if and only if the two following conditions are met:

- The payoff of any player i belonging to coalition C depends only on the players present in C .
- The coalitions form according to the preferences relation of all players.

A **round** is defined as a complete tour of all players. In other words, in a round, each player plays exactly once.

B. Hedonic coalition formation

Data gathering by mobile sinks is modeled as a hedonic coalitional game with n rational players, where n is the number of collect points to visit. The set of players is denoted

$\{i, i = 1 \dots n\}$, where i is a collect point to be visited by a mobile sink.

The coalitional game ensures that at any time, any player belongs to exactly one coalition. Each coalition represents the set of collect points visited by a same mobile sink.

Each player plays in sequence. Let \mathcal{P} denote the current partition formed by the coalitions existing when any player i is playing. Player i tries to strictly increase its payoff by joining another coalition in $\mathcal{P} \cup \{\emptyset\}$.

In this hedonic coalition game, each player i applies the switching rule to increase its payoff as follows:

Switching rule: Any player i leaves its current coalition C_1 to join coalition $C_2 \in \mathcal{P} \cup \{\emptyset\}$ if and only if

$$C_2 \cup \{i\} >_i C_1,$$

or equivalently

$$u_i(C_2 \cup \{i\}) > u_i(C_1).$$

Notice that each player is selfish: it leaves a coalition to join another independently of the effects of this move on the other players.

More precisely, the payoff $u_i(C)$ of any player i belonging to coalition C is defined by Algorithm 1. We notice that the payoff of player i is set to $-\infty$ if either i selects a coalition already in its history or if the tour duration violates the constraint $Lmax$.

Algorithm 1 Payoff of player $i \in C$

```

if  $C \in history(i)$  then
   $u_i(C) = -\infty$ 
else
  if  $2 * tour\ duration > Lmax$  then
     $u_i(C) = -\infty$ 
  else
    if  $size(C) = 1$  then
       $u_i(C) = -maxdistance - n - 1$ 
    else
       $u_i(C) = -max_{i \in C} distance(i, centroid(C)) - n/size(C)$ 
    end if
    insert  $C$  in  $history(i)$ 
  end if
end if

```

where:

- $Lmax$ is the maximum latency acceptable by the application. The value of this parameter may also take into account the residual energy of the robot. Knowing the power of the mobile sink, the residual energy of the mobile sink can easily be translated into a maximum tour duration. To meet the latency constraint, the tour duration of each mobile sink must be less than $Lmax/2$.
- $centroid(C)$ denotes the centroid of the collect points belonging to the coalition C . A player is discouraged to belong to a coalition whose centroid is far from itself. This player would increase the maximum distance of coalition members to the centroid, denoted

by $max_{i \in C} distance(i, centroid(C))$, leading to a poor payoff.

- $size(C)$ denotes the size of the coalition (i.e. the number of its members). $n/size(C)$ denotes the number of coalitions that would be obtained if all coalitions had the size of coalition C .
- $maxdistance$ denotes the maximum distance between two collect points. As a consequence the value $-maxdistance - n - 1$ is a value that is never reached by a collect point belonging to a coalition whose size is strictly higher than 1.

The payoff of any player is computed in order to favor coalitions that collect a large amount of data while minimizing the tour duration. Each player is strongly discouraged to join again a coalition that it previously left. In addition, the payoff of a player in a coalition increases with the size of data collected until a limit given by the latency constraint.

Theorem 1: This hedonic coalitional game converges towards a Nash stable and individually stable partition, whatever the initial partition.

Proof 1: The history of this hedonic coalitional game is an ordered sequence of partitions starting from the initial partition $\mathcal{P}_0: \mathcal{P}_0 \rightarrow \mathcal{P}_1 \dots \mathcal{P}_k \rightarrow \mathcal{P}_{k+1} \dots$. According to the definition of a hedonic coalition game, there is a transition of partition \mathcal{P}_k to partition \mathcal{P}_{k+1} if and only if there is a player i that strictly increases its payoff by moving from its current coalition in \mathcal{P}_k to a coalition in partition \mathcal{P}_{k+1} . Since the size of \mathcal{C}_i , the set of all possible coalitions containing i , is finite, no player moves back to a coalition already visited and the number of players is finite, hence the history is finite. Hence, a final partition is obtained.

We now prove that no player can get a strictly higher payoff than this obtained in the final partition, even in joining another coalition. By contradiction, let us assume there is a player i that belongs to coalition C in the final partition such that i would strictly increase its payoff by joining another coalition C' . In other words, $u_i(C' \cup \{i\}) > u_i(C)$. According to the switching rule, player i would move to C' . Hence, the partition considered is not the final one: a contradiction.

C. Advantages of the coalition formation modeling

Modeling the data gathering problem as a coalitional game allows a very simple expression of the constraints where:

- Each collect point is visited by exactly one mobile sink.
- Each mobile sink visits at least one collect point.

The coalition formation problem is solved in an elegant way. The final partition obtained is such that:

- The number of coalitions in the final partition provides the number of mobile sinks needed by data gathering.
- Each coalition belonging to the final partition represents the set of collect points visited by a same robot.

There is no parameter to tune and no risk of obtaining a bad solution because of an inadequate parameter tuning.

Notice however that the sequence of visits performed by the mobile sink is an ordered permutation of the collect points

belonging to the coalition. The permutation chosen is the one minimizing the tour duration of the mobile sink considered.

D. Discussion

We can notice that if the acceptable latency L_{max} is higher than twice the tour duration of a mobile sink visiting all collect points, the game becomes super-additive and the grand coalition is obtained as the final solution: it is the coalition with the highest payoff.

The proof of Theorem 1 suggests that the final partition obtained depends both on the initial partition and on the order according to which the players play. This is corroborated by our simulation results. We observe that the possibilities for a player to join a coalition improving its payoff decrease with the time because the other players having played before i have already joined the interesting coalitions and the number of members in a coalition is ruled by the latency constraint as we will see in the next section.

Let us focus on how to implement the coalition formation game in a network where the players (i.e. the collect points) are disconnected. To perform data gathering according to our solution, a number of mobile sinks should be provided. Before starting the data gathering phase, we need to use one of these mobile sinks as a mobile node (e.g. robot) in charge of visiting each collect point successively. During its visit, the mobile node brings the game to allow the collect point to play. The game is then updated and the mobile node moves to the next player (i.e. the next collect point). When the game ends, a number of mobile nodes equal to the number of coalitions formed become mobile sinks. Each mobile sink visits all the collect points belonging to the coalition it is in charge of.

Each time the problem is changed, new coalitions must be formed. The process of selecting a mobile node to bring the game to each player is repeated and a new game is played.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the coalitional game in terms of the number of coalitions formed, the tour duration as well as the complexity of the game expressed as the number of rounds needed to reach the final stable partition and the total number of switches.

A. Simulation parameters

Initially, each player is alone in its coalition. In other words the initial partition in the coalitional game is given by $\{\{1\}, \{2\}, \dots, \{n\}\}$. In all simulation runs, each player i computes its payoff when joining each coalition existing when it is playing; instead of joining the first coalition that increases its payoff, it only joins the coalition that gives the highest payoff. This policy is called "Best coalition" and it is used instead of "First coalition".

Each simulation is defined by its parameters such as the number and the positions of the collect points in the 2D area considered, the value of L_{max} , the speed of the mobile sinks. The value of these parameters are given in Table I.

Notice that, for each configuration (i.e. a given number of collect points), we performed 30 simulation runs conducted

TABLE I: Simulation parameters

L_{max}	3000s
L_{speed}	1m/s
Area	$500\sqrt{2}m \times 500\sqrt{2}m$
Number of collect points	$n \in \{20, 40, 60, 80, 100\}$
Position of collect points	randomly chosen in the area
Simulation result	average of 30 simulations

using a desktop computer Intel Xeon E5 1620 processor with 8-Core 3.6GHz and 8 Gb of memory.

B. Impact of the ordered sequence of play

In this series of simulations, we evaluate the impact of the order of play on both: the average number of coalitions obtained (i.e. number of mobile sinks) and the maximum tour duration. We consider three orders: random, rich-to-poor and poor-to-rich.

With the random order, the play order is selected randomly at the beginning of the game and is used during the whole game. With the rich-to-poor order, at the beginning of each round r the players are sorted according to the decreasing payoff order and they play in round r according to this order. Whereas with the poor-to-rich order, they are sorted according to the reverse order.

Let us consider the simple example of 20 collect points and play the game once with the poor-to-rich order, rich-to-poor order and random order. Results are illustrated in Figures 1, 2 and 3, respectively. We can see that the order has an impact on the number of coalitions. In this example, the poor-to-rich order tends to decrease the number of coalitions obtained compared to the random order.

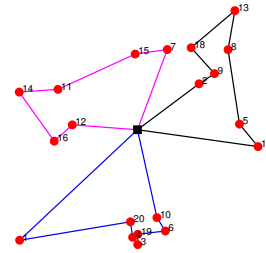


Fig. 1: Poor-to-rich order with 20 collect points.

Figure 4 depicts the average number of coalitions obtained as a function of the number of collect points. As expected, the number of coalitions increases with the number of collect points in the three orders considered, because of latency constraint. We also observe that with small configurations (e.g. 20 and 40 collect points) the order in which the players play has no visible impact on the number of coalitions formed. For each order (i.e. poor-to-rich, rich-to-poor and random), there is a configuration where it provides the smallest number of coalitions.

Figure 5 shows the maximum tour duration as a function of the number of collect points. We observe that the results

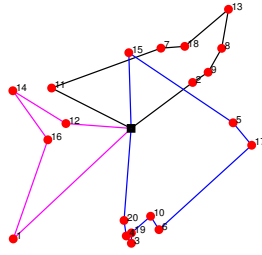


Fig. 2: Rich-to-poor order with 20 collect points.

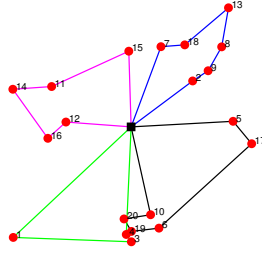


Fig. 3: Random order with 20 collect points.

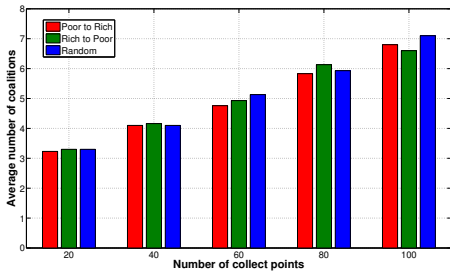


Fig. 4: The average number of coalitions as a function of the number of collect points.

obtained are very similar for the three orders. This is due to the latency constraint that limits the tour duration mainly for large configurations.

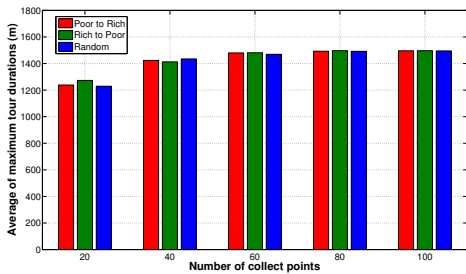


Fig. 5: The maximum tour duration as a function of the number of collect points.

C. Complexity results

We define the complexity of this coalition game as the number of rounds needed to obtain the final partition. This number of rounds is defined as the number of times each player played. It is also interesting to evaluate the total number of switches that have been done by the players to reach the final partition.

Figure 6 illustrates the average number of rounds as a function of the number of collect points. Whatever the number of collect points tested less than or equal to 100, the number of rounds is less than 14. The poor-to-rich order provides the best results in terms of number of rounds when the number of collect points is less than 60.

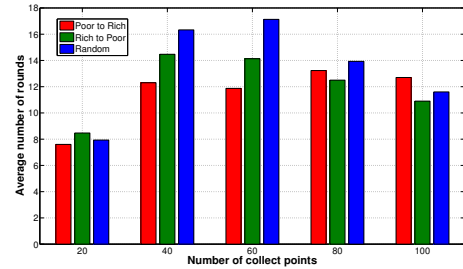


Fig. 6: The average number of rounds as a function of the number of collect points.

Figure 7 depicts the average number of switches as a function of the number of collect points. The average number of switches is ≤ 200 for configurations with a number of collect points in the interval $[20, 100]$. The rich-to-poor order provides the smallest number of switches. This can be explained by the fact that the player which has the highest payoff plays first. As a consequence, it can join the best coalition. The latency constraint tends to prohibit the other players to join this best coalition.

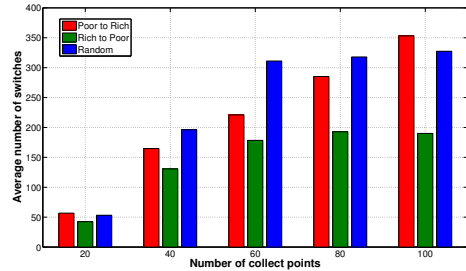


Fig. 7: The average number of switches as a function of the number of collect points.

How long does it take to complete the game with the formation of the final partition? Figure 8 shows the average simulation time as a function of the number of collect points.

In all cases, we can conclude that the convergence of the coalition game is very fast: it requires only a few number of rounds (i.e. less than 14) and less than 18 minutes to obtain the final partition. Since this result holds for a high number of collect points, we conclude that the coalition game is scalable.

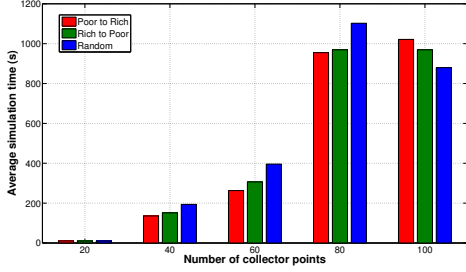


Fig. 8: The average simulation time as a function of the number of collect points.

In Section VI, we will compare the results obtained by the coalitional game with an optimization heuristic based on the genetic approach.

V. EXTENSION: HOW TO SUPPORT DIFFERENT LATENCY REQUIREMENTS

We now assume that all data to gather may have various latencies and each of them has to be delivered to the static sink by a mobile sink before its deadline that is its latency. For the sake of simplicity, we consider three latencies L_1^{max} , L_2^{max} and L_3^{max} , with $L_1^{max} < L_2^{max} < L_3^{max}$. We now see how to model this problem as a coalition game.

A. An extended payoff function

The problem is modeled as a coalition game with n players that are the collect points to visit. These players will group in coalitions in order to maximize their payoffs. The payoff function is updated in order to take into account the existence of several latencies. In addition, a coalition must meet the latency of the member having the strongest latency constraint. In other words, any coalition C must meet latency:

- L_1^{max} if C contains at least one member with latency L_{max1} ,
- L_2^{max} if C contains no member with latency L_1^{max} but at least one member with latency L_2^{max} ,
- L_3^{max} if C contains no member with latency L_1^{max} or L_2^{max} .

Let us denote by $L1$ the set of collect points with latency L_1^{max} , $L2$ the set of collect points with latency L_2^{max} and $L3$ the set of the remaining collect points.

The payoff $u_i(C)$ of any player i belonging to coalition C is defined as:

B. Simulation results

We consider three latencies: $L_1^{max} = 1000s$, $L_2^{max} = 1500s$, $L_3^{max} = 3000s$ and we define for each L_i^{max} the group of collect points L_i with this latency:

- $L1 = \{1, 13, 10, 8, 11, 19, 15\}$
- $L2 = \{14, 9, 3, 6, 20, 18\}$
- $L3 = \{5, 17, 7, 2, 16, 4, 12\}$

Algorithm 2 Extended Payoff of player $i \in C$

```

if  $C \in history(i)$  then
   $u_i(C) = -\infty$ 
else
  if  $C \cap L1 \neq \emptyset$  and  $2 * tour\ duration > L_1^{max}$  then
     $u_i(C) = -\infty$ 
  else
    if  $C \cap L2 \neq \emptyset$  and  $2 * tour\ duration > L_2^{max}$  then
       $u_i(C) = -\infty$ 
    else
      if  $2 * tour\ duration > L_3^{max}$  then
         $u_i(C) = -\infty$ 
      else
        if  $size(C) > 1$  then
           $u_i(C) = -max_{i \in C} distance(i, centroid(C)) - n / size(C)$ 
        else
           $u_i(C) = -max\ distance - n - 1$ 
        end if
      end if
    end if
  end if
end if

```

Figure 9a presents the simulation result obtained for 20 collect points and the three latencies L_1^{max} , L_2^{max} and L_3^{max} . To satisfy the latency constraints, 5 mobile sinks are needed to collect data. Table II illustrates the tour duration and the corresponding latency matching each coalition.

Notice that with only one latency equal to the highest latency L_3^{max} , 3 mobile sinks are sufficient to report the data of these 20 collect points as depicted in Figure 9b. However, no mobile sink meets the smallest latency L_{max1} . Furthermore, considering only the smallest latency L_1^{max} would lead to a number of mobile sinks much higher than required to meet the three latency constraints L_1^{max} , L_2^{max} and L_3^{max} .

As a conclusion, the latency constraint has a big impact on the number of coalition formed and then on the number of mobile sinks needed. A coalition game able to deal with multiple latencies is much more representative of real situations. In addition, it finds the smallest number of mobile sinks meeting the latency constraints expressed by the application considered.

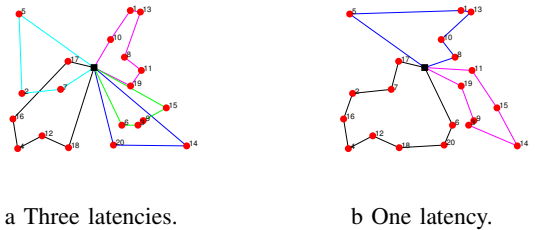


Fig. 9: Poor-to-rich order.

TABLE II: Results corresponding to Figure 9a

Coalition	[16 12 18 17 4]	[8 1 11 10 13 19]	[20 14]	[3 9 6 15]	[5 2 7]
Duration(s)	997,48	745,16	987,51	706,05	901,44
Latency	L_2^{max}	L_1^{max}	L_2^{max}	L_1^{max}	L_3^{max}

VI. COMPARATIVE EVALUATION

In this section we want to compare the solution based on game theory with a solution found by an heuristic. We first define the problem as an optimization problem that we solve with a centralized algorithm based on genetics.

A. Optimization Problem

Let $G = (V, E)$ be the directed graph, where, V , (i.e. vertices) is the set of collect points in the area considered A , and E , (i.e. edges) is the possible movement followed by any mobile sink in order to move from one collect point to another.

Let $C = (c_{ij})$ be a cost matrix, denoting the cost for a mobile sink when traveling from vertex i to vertex j . We consider that $c_{ij} = d_{ij}/l_s$, where d_{ij} is the euclidean distance required for traveling from collect point i to collect point j , i and $j \in V$ and l_s is the linear speed of each mobile sink. Finally, we assume that all the mobile sinks are located at a given place in A considered as the depot and all the mobile sinks start and end their tour at that depot. Our optimization problem is defined as follows:

$$\min \max_{k \in K} \sum_{i \in A} \sum_{j \in A} \frac{d_{ij}}{l_s} x_{ij}^k \quad (1a)$$

$$\max \frac{(\sum_{k \in K} \sum_{i \in A} \sum_{j \in A} \frac{d_{ij}}{l_s} x_{ij}^k)^2}{|K| \sum_{k \in K} (\sum_{i \in A} \sum_{j \in A} \frac{d_{ij}}{l_s} x_{ij}^k)^2} \quad (1b)$$

$$\text{s.t.} \quad \sum_{i \in A} \sum_{k \in K} x_{ij}^k = 1, \forall j \in A \quad (1c)$$

$$\sum_{i \in A} x_{ip}^k - \sum_{i \in A} x_{pj}^k = 0, \forall p \in A; \forall k \in K \quad (1d)$$

$$\sum_{j \in A} x_{oj}^k = 1, \forall k \in K \quad (1e)$$

$$\sum_{j \in A} x_{jo}^k = 1, \forall k \in K \quad (1f)$$

$$u_i^k - u_j^k + n x_{ij}^k \leq n - 1, \forall i \neq j \in A \setminus \{o\}; \quad (1g)$$

$$\forall k \in K \quad (1g)$$

$$x_{ij}^k \in \{0, 1\}, \forall i, j \in A; \forall k \in K \quad (1h)$$

where x_{ij}^k is a binary variable that is equal to 1 if the mobile sink k moves directly from the collect point i to the collect point j , and 0 otherwise. $|K|$ denotes the cardinal of the set K .

As indicated in the problem formalization, the first objective (1a) is to minimize the tour duration for each mobile sink. The second objective (1b) maximizes the fairness between the mobile sinks in terms of tour duration. The first constraint (1c) states that each link should be used by exactly one mobile sink. The next constraint (1d) is for flow conservation. More precisely, this constraint guarantees that when a mobile sink visits collect point i , it must also depart from this collect point. The constraint (1e) imposes the requirement that the number of mobile sinks leaving the depot should not exceed the maximum number of mobile sinks at the depot, and (1f) further

imposes the number of mobile sinks that leaves the depot should be identical to the number of mobile sinks returning to the depot. Finally, the constraint (1g) is used to prevent subtours. This constraint is known as a *subtour elimination constraint* (SECs). The literature is rich with several proposals for SECs. In our formulation, we choose the one originally proposed by Miller et al. [14] noted as MTZ-SECs. Basically, u_i^k denotes the number of collect points visited on the path selected by mobile sink k from the depot to the collect point i . This constraint imposes both capacity and connectivity requirements. Indeed, for all k , when $x_{i,j}^k = 0$, constraint (1g) is relevant since $u_i^k \leq n - 1$ and if when $x_{i,j}^k = 1$, it imposes that $u_j^k \geq u_i^k + 1$.

Unfortunately, given the important number of variables in our problem formulation it may not be practical to directly solve our problem, and even for small size systems. One possible approach is to use heuristic approaches such as Genetic Algorithm.

B. Heuristic solution: Genetic approach

In our optimization problem, more than one objective function is involved. The task of finding one or more optimal solutions is known as multi-objective optimization. To solve this multi-objective optimization problem we decide to use the NSGA-II [15] algorithm based on the genetic approach. NSGA-II, for Non dominated Sorting Genetic Algorithm, is used to solve multi-objective optimization problems.

NSGA-II begins with an initial population P of candidate solutions called individuals. Each population is called generation. At each generation, the fitness, which is the value of the objective function of every individual in the population, is evaluated in order to select the best individuals. At each generation, applying the crossover and mutation operators on individuals from the population generates new individuals. Hence, an auxiliary population Q is formed. Both the current P and the new population Q are merged together to form one set of solutions R , which is sorted according to the non-domination and crowded comparison. Finally, only the best individuals in R are included in the next generation and participate in the production step while the other individuals are deleted. These steps are repeated until the maximum number of iterations is reached.

To solve our problem using the genetic algorithm, we adopt both the genetic representation of a solution (i.e. individual), the operators (crossover and mutation) and the fitness function (i.e. evaluating the satisfaction degree of the objectives) that are detailed in [16].

C. Comparison between genetic and game theory

We consider three configurations of 10, 15 and 20 collect points respectively. For each of them we compare the results obtained with game theory and genetic algorithms. Simulation results show that both approaches obtain very close results in terms of maximum tour duration and fairness. However, game theory outperforms the genetic approach in terms of

execution time: for 20 collect points it requires 327.69 times less execution time than genetic algorithm.

TABLE III: Comparison between Game Theory based solution and Genetic approach based solution

NB. collect points	Max tour duration		Fairness		Time(s)	
	Genetic	Game theory	Genetic	Game theory	Genetic	Game theory
10	837.36	837.36	0.999	0.999	11.01	7
15	1214	1214	0.997	0.997	596.9	2.5
20	1154.78	1219.66	0.999	0.999	4260.8	13.0
40	-	1227.0	-	0.997	-	99.2
60	-	1470.0	-	0.997	-	331.0
80	-	1496.87	-	0.999	-	698.0
100	-	1497.06	-	0.995	-	846.47

Both the coalition game and the genetic approach are centralized approaches. We now evaluate the ratio of the coalition game solution over the genetic solution for three different criteria: the maximum tour duration, the fairness between the tour duration of the mobile sinks and the time needed to get the solution. These ratios are given in Table IV. For small configurations (i.e. up to 15 collect points), we obtain the ideal ratio of 1 with regard to both the maximum tour duration and fairness. When the number of collect points increases, this ratio increases, but very moderately: it is only 1.05 for 20 collect points. However, if we consider the time needed to obtain the final solution, the coalition game is very advantageous.

TABLE IV: Comparative evaluation

NB. collect points	Max tour duration ratio	Fairness ratio	Time ratio
10	1	1	0.64
15	1	1	0.004
20	1.05	1	0.08

When the number of collect points is strictly higher than 20, the genetic algorithm needs more than two days to provide the results. This is due to a necessary increase in the population size and in the number of rounds used to improve the population. That is why, we did not compare the solution based on game theory with the solution based on genetics when the number of collect points is high.

VII. CONCLUSION

In wireless sensor networks, sensor nodes have two major tasks: the first one is the detection of events and the second one is the report of the associated data to the sink. However, sensor nodes may be disconnected from the sink, then the second task cannot be satisfied. In such a case, one or multiple mobile sinks should visit the static sensor nodes (considered as collect points) to collect data. However, some constraints should met, such as the latency of data delivery, the energy of the mobile sinks (which are battery equipped), and the limited number of these mobile sinks. In this paper, we proposed a game theory based approach to determine the number of mobile sinks and the set of collect points that should be visited by each mobile sink, while meeting these constraints. Our game is a coalition

formation game that optimizes the maximum tour duration and maximizes the fairness in term of tour duration of mobile sinks. In this game, the players are the collect points and each coalition formed corresponds to a tour of a mobile sink. Simulation results show that the order in which the players play has an impact on the number of coalitions formed and on the maximum tour duration. In addition, the convergence of the coalition game to a final partition is very fast in terms of number of rounds and simulation times, even when the number of collect points is great (e.g. 100 collect points). We also performed a comparison evaluation with an optimization heuristic based on genetic algorithms. In small configurations, both centralized solutions provide similar results. When the number of collect points increases the optimization solution may provide better results than the game theory but at the cost of a very long execution time. As a conclusion, our solution based on a coalition formation game is a scalable solution that needs short execution time to provide very good results.

REFERENCES

- [1] Z. Vincze, D. Vass, R. Vida, A. Vidacs, A. Telcs, *Adaptive Sink Mobility in Event-driven Multi-hop Wireless Sensor Networks*. InterSense, 2006.
- [2] *Supple: A Flexible Probabilistic Data Dissemination Protocol for Wireless Sensor Networks*
- [3] Prolonging Network Lifetime in Wireless Sensor Networks with Path-Constrained Mobile Sink
- [4] Mobile Element Scheduling for Periodic Sensor applications
- [5] G. Laporte *The vehicle routing problem: an overview of exact and approximate algorithms* European Journal of Operational Research, 59 (1992), pp. 345358
- [6] M. J. Osborne, A. Rubinstein. *A course in game theory*. MIT press, 1994.
- [7] T. Basar, G. J. Olsder. *Dynamic noncooperative game theory*. Vol. 200. London: Academic press, 1995.
- [8] O. Ait Oualhaj, A. Kobbane, M. Elmachour, E. Sabir, J. Ben-Othman. *A coalitional-game-based incentive mechanism for content caching in heterogeneous Delay Tolerant Networks*. In 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 987-992, 2015.
- [9] W. Saad, H. Zu, T. Basar, M. Debbah, A. Hjrungnesng, *Coalition formation games for collaborative spectrum sensing*, IEEE Trans. on Vehicular Technology, vol. 60, no. 1, 2011.
- [10] W. Saad, H. Zu, T. Basar, M. Debbah, A. Hjrungnesng, *Hedonic coalition formation for distributed task allocation among wireless agents*, IEEE Trans. on Mobile Computing, vol. 10, no. 9, Sep. 2011.
- [11] D. Niyato, P. Wang, H. Tan, W. Saad, D. Kim. Cooperation in delay tolerant networks with wireless energy transfer: Performance analysis and optimization. IEEE Transactions on Vehicular Technology, Vol. 64 , no. 8, 2015.
- [12] N. Ghosh, I. Banerjee, T. Samanta. Energy efficient coverage of static sensor nodes deciding on mobile sink movements using game theory. In Applications and Innovations in Mobile Computing (AIMoC), pp. 118-125, 2014.
- [13] I. Khoufi, P. Minet, M-A. Koulali and M. Erradi *A game theory-based approach for robots deploying wireless sensor nodes*, Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, Croatia, 2015.
- [14] C. E. Miller, A. W. Tucker and R. A. Zemlin *Integer Programming Formulation of Traveling Salesman Problems* Journal of the ACM, Vol. 7 I. 4, pp. 326-329, 1960.
- [15] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan *A fast and elitist multi-objective genetic algorithm: NSGA-II* IEEE Transactions on Evolutionary Computation, Vol. 6, I. 2, 2002.
- [16] I. Khoufi, M. Hadded, P. Minet and A. Laouiti *Optimized Trajectories of Multi-Robot Deploying Wireless Sensor Nodes* IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Abu-Dhabi, United Arab Emirates, 2015.