Unmanned Aerial Vehicles Path Planning for Area Monitoring

Ines Khoufi, Pascale Minet, Nadjib Achir

To cite this version:

Ines Khoufi, Pascale Minet, Nadjib Achir. Unmanned Aerial Vehicles Path Planning for Area Monitoring. PEMWN 2016 : The 5th IFIP International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks, Nov 2016, paris, France. <hal-01410071>

HAL Id: hal-01410071
https://hal.inria.fr/hal-01410071

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.

Unmanned Aerial Vehicles Path Planning for Area Monitoring

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

Nadjib Achir
Universite Paris 13, Sorbonne Paris Cite – L2TI,
99 Avenue J-B Clément, 93430 Villetaneuse, France.
Email: nadjib.achir@univ-paris13.fr

***Abstract*—In this paper we are interested in area monitoring using Unmanned Aerial Vehicles (UAVs). Basically, we propose a path planning approach for area monitoring where UAVs are considered as mobile collectors. The area to be monitored is divided into cells. The goal is to determine the path of each UAV such that each cell is covered by exactly one UAV, fairness is ensured in terms of the number of cells visited by each UAV and the path of each UAV is minimized. To meet our goal, we proceed in two steps. In the first step, we assign to each UAV the cells to visit. In the second step, we optimize the path of each UAV visiting its cells. For the first step, we propose two solutions. The first solution is based on cluster formation, each cluster is made up of the set of cells monitored by a same UAV. The second solution is based on game theory and uses coalition formation to determine the cells to be monitored by each UAV. In the second step and for both solutions, we propose to apply optimization techniques to minimize the path of each UAV that visits all its cells.**

I. INTRODUCTION

The Unmanned Aerial Vehicles (UAVs) industry is projected to grow considerably in the near term along with the range of commercial capabilities. Current UAVs go from advanced technology requiring high technical skills and expertise to simple ones offering more flexibility and simplicity of use, thus, allowing a wide spectrum of mission profiles. UAVs were initially adopted by military for missions that are considered dangerous. Moreover, the absence of an aircrew means that a great deal of space and weight can be saved. Now, UAVs can perform increasingly sophisticated missions due to their small size, discretion and accuracy of observation. The development of unmanned aerial vehicles over a wide range from small UAVs on infantryman scale, to the strategic UAVs of high technology, encourages armed forces to integrate them gradually into the panoply of air assets involved in operational theaters.

Although the market is almost nonexistent today, this is most likely in the civil field that UAVs are expected to play the largest role, and that due to the flexibility and versatility of their use. The range of potential applications is almost unlimited. UAVs are being the more credible to meet the need that is not covered by manned aircraft. This is the case of missions that can be considered dangerous or physically painful for the crew. As a first approach, by simple effect of imitation, civilians may observe the culture of UAVs among military, and its gradual extension outside the security and defense use. Many civilian uses also remain to be discovered and, with them, the need for equipment and specialized software.

The use of UAVs in commercial applications is expected to expand in a number of areas. Some of the currently proposed civil and commercial applications of UAVs include security awareness, disaster response including search and support to rescuers, critical infrastructure monitoring, etc. [1][2].

In this paper, we are interested in area monitoring using UAVs. One classical approach for area monitoring is to use mobile robots. The goal is to simply collect information from this area and report it to the central point. Unfortunately, mobile robots may fail to monitor the whole area since obstacles may exist. These obstacles may prohibit mobile robots to reach some points. However, UAVs could be considered as a serious alternative for area monitoring. Indeed, the use of UAVs will make the monitoring much easier: first there is no need to avoid obstacles, which mean fewer constraints on the UAV trajectories. Second, since the sensing range may cover a large area, the monitoring will be faster. Third, a few number of UAVs will be sufficient to monitor the large area.

II. RELATED WORK

In this section, we describe some studies related to area monitoring using UAVs. Most of the existing works considered mobile robots, and focuses mainly on path planning problem. The most famous path planning problem is the Traveling Salesman Problem, TSP, [3], [4]. It has been proved NP-hard in [5]. A formalization of this problem is given in [3]. Some enhancements have been brought in [4] to avoid subtours. Recently, this problem has been generalized to m-TSP with multiple traveling salesmen. An overview of formulations and solutions is given in [6]. The m-TSP problem has many real-life applications such as school bus routing, workshop scheduling, technical crew scheduling, transportation planning in smart cities, truckload pickup and delivery, etc. The literature provides solutions for TSP and m-TSP. Near-optimal solutions can be obtained with genetic [7], [8]. Some of them [7] are applied in real applications such as iron and steel industries and the production improvements obtained are considerable.

In [9], the path of a single mobile robot deploying wireless sensor nodes and placing them at some given positions (i.e. those of points of interest) is minimized by selecting turn points of the robot. Another robot path is found to collect data from these sensor nodes by minimizing the number of stops done by the robot, taking advantage of the proximity of some sensor nodes to upload their data without moving like in [10], where however multiple robots are considered. The solution proposed did not address fairness among robots tour duration, unlike ours.

In [11], we defined the MRDS problem where multiple robots are in charge of deploying wireless sensor nodes and placing them at some given positions called points of interest. The problem addressed in this paper differs from the MRDS problem by the objectives considered. The fairness is expressed here by the Jain's index. The computation of the flight duration of a UAV differs (no rotation time). Furthermore, in this paper, the number of UAVs is given and all of them are used unlike in the MRDS problem where the number of robots used is minimized.

The originality of this contribution is the simplicity of the solutions proposed. The processing time is lower than this obtained by genetic algorithm [11] because the space of solutions explored in reduced.

III. System model

In this paper, we focus on monitoring applications that require full coverage of the area to be monitored. Permanent connectivity is not needed to meet the latency requirements. Temporary connectivity is provided by UAVs visiting these points to collect their data. The latency requirements are met by any UAV visiting at most $\lceil \frac{Number\ of\ points\ to\ visit}{Number\ of\ UAVs} \rceil$. That is why we do no longer consider latency requirements but we focus on the fairness between UAVs in terms of the number of points visited.

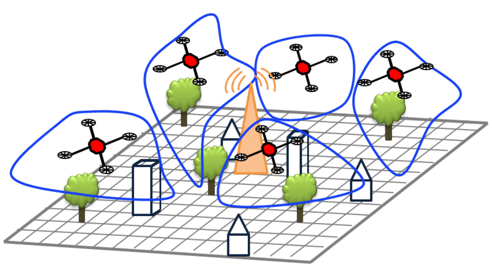

Fig. 1: Area monitoring using UAVs

In order to ensure full area coverage we propose to apply the optimal deployment on this area that corresponds to the triangular tessellation. The optimal deployment ensures full coverage with the minimum number of sensor nodes. Then, each sensor node has 6 neighbors. Each node ensures the coverage of an hexagonal cell.

Let A be the 2D-geographical rectangular area to be monitored. Let n be the number of cells in A. Let u be the number of UAVs. In this paper, the number of UAVs is set to 4.

Our first goal is to select the set of cells to be visited by each UAV provided that each cell should be visited exactly once by one of the UAVs. The second goal is to optimize the tour duration of each UAV in charge of monitoring the area A. For that, we apply the 2-Opt [12] heuristic to optimize the UAV path's.

To meet our goal, we proceed in two steps. In the first step, called cell assignment, we assign to each UAV the cells to visit. In the second step, called path optimization, we optimize the path of each UAV visiting its cells. For the cell assignment, we propose two solutions, one based on cluster formation and the other based on coalition formation. Path optimization is done using 2-Opt algorithm [12].

Figure 2 presents a solution for 4 UAVs to monitor an 200m*200m Area.

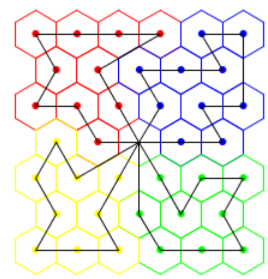

Fig. 2: Path planning Fro 4 UAVs

IV. First solution : Cell assignment by Cluster formation

In this section, we present a simple algorithm to select for each UAV the set of cells to monitor. This algorithm proceeds in iteration. At each iteration, each UAV selects a cell to be monitored. The selection of the cell depends on :

- The free cells : cells which have not yet been selected by UAVs.
- The cells already selected by this UAV. To minimize its path, any UAV is only allowed to choose a cell, among the free cells, that is adjacent to its already selected cells.
- The distance of the candidate cell to all cells selected by the other UAVs. The idea is to choose the farthest cell to the cells selected by other UAVs in order to ensure fairness between UAVs.

Algorithm 1 presents the algorithm used to build the clusters corresponding to the set of cells visited by each UAV. This algorithm uses the principles previously given. A result obtained by this algorithm in an area of $200m * 200m$ is depicted in Figure 2.

V. Second solution : Cell assignment by Coalition formation

A. Notations and definitions

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

Algorithm 1 Cluster formation

$Grid\_List$ the set of cells to be visited including the sink cell
$Cluster\{i\}$ = a given cell adjacent to the sink cell
$Free\_cells = Grid\_List - \{Sink, Cluster\{i\}\}$
while $Free\_cells \neq \emptyset$ **do**
  $All\_Chosen\_Cells = Cluster\{1\} \cup Cluster\{2\} \cup Cluster\{3\} \cup Cluster\{4\}$
  $Candidate\{i\}$ = all free cells adjacent to $Cluster\{i\}$
  tab = Table of distances initialized to $+\infty$
  for $j = 0$ to $Candidate\{i\}.size$ **do**
    for $k = 0$ to $size(All\_Chosen\_Cells)$ **do**
      $Dist = \min Distance(Candidate\{i\}[j], All\_Chosen\_Cells[k])$
      $tab[k] = Dist$
    end for
    $max = tab[0]$
    $index = 0$
    for $k = 1$ to $size(tab)$ **do**
      $max = \max tab[k]$
      $index = k$
    end for
    $cell = Candidate\{i\}[index]$
    $Cluster\{i\} = Cluster\{i\} \cup cell$
    $remove\ cell\ from\ Free\_cells$
  end for
end while

For any player $j \in [1, n]$, let $u_j(C)$ be the payoff of player j when belonging to coalition C.

According to [13], 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.
- According to the preference relation: each player always prefers to belong to the coalition increasing its payoff.

B. Hedonic coalition formation

Data gathering by UAVs 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 $\{j, j = 1 \ldots n\}$, where j is a cell to be visited by a UAV.

The coalitional game ensures that at any time, any player belongs to exactly one coalition. Each coalition represents the set of cells visited by a same UAV. Initially, each cell forms a coalition reduced to itself. There are 4 particular coalitions in $\mathcal{C}_{uav}$, one per UAV that can be joined by the cells. Each of these 4 coalitions contains a cell adjacent to the sink cell that is the initial position of the UAV.

Each player plays in sequence. Let $\mathcal{C}_{uav}$ denote the coalition formed by the UAVs when any player j is playing. Player j in a coalition C tries to strictly increase its payoff by joining another coalition $C' \in \mathcal{C}_{uav}$, provided that j is adjacent to at least one cell in coalition C'.

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 to join coalition $C' \in \mathcal{C}_{uav}$ if and only if

$$u_j(C' \bigcup \{j\}) > u_j(C)$$

and j is adjacent to at least one cell in coalition C'.

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 $u_j(C)$ of any player j belonging to coalition C is defined as:

Algorithm 2 Payoff of player $j \in C$

if $size(C) == 1$**or**$C \in History\{j\}$ **then**
  $u_j(C) = -\infty$
else
  if $(size(C) == \lfloor \frac{n}{u} \rfloor)$ **or** $(size(C) == \lceil \frac{n}{u} \rceil)$ **or** $(size(C) == (\lceil \frac{n}{u} \rceil + 1))$ **then**
    $u_j(C) = \lfloor \frac{n}{u} \rfloor$
  else
    $u_j(C) = 0$
  end if
  if $(centroid(C) \leq \frac{\sqrt{L^2+W^2}}{4})$ **then**
    $u_j(C) = u_j(C) + \lfloor \frac{n}{2*u} \rfloor$
  end if
end if

- $centroid(C)$ denotes the centroid of the cells 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 $max_{j \in C} distance(j, centroid(C))$, leading to a poor payoff.
- $size(C)$ denotes the size of the coalition (i.e. the number of its members).
- $History\{j\}$ denotes the set of coalitions that j left.
- L and W are the length and the width of the area to be monitored.

The payoff of any player is computed in order to favor the fairness between coalitions and coalitions made up of close cells. Each player is strongly discouraged to join again a coalition that it previously left.

VI. Comparative Evaluation

In this section, we compare the results obtained by the cluster formation and the coalition game, both followed by path optimization, in terms of the number of cells visited by each UAV, fairness, average distance traveled by an UAV and processing time.

In order to evaluate the fairness between UAVs, we compute the Jain's fairness index:

$$\frac{\sum_{i=1}^{u} x_i}{u * \sum_{i=1}^{u} x_i^2} \tag{1}$$

We apply this index to :

- the number of cells visited by each UAV, in such a case x_i denotes the number of cells visited by UAV_i.
- the distance traveled by each UAV, in such a case x_i denotes the distance traveled by UAV_i.

We consider, 4 UAVs and 5 configurations where the dimension of the area to be monitored, $L = W$ ranges from $200m$ to $400m$ by step of $50m$.

Assuming a sensing range of $20m$ which corresponds to the hexagon radius to get a number of hexagonal cells ranging from 39 to 161.

All our experiments were conducted using a desktop computer Intel (R) E7 5600U processor with 8-Core 2.6GHz and 16 Gb of memory.

Figures 3 and 4 depict the results obtained by the cluster formation and coalition game, respectively, for a monitored area of $300m * 300m$. In this configuration, the sink occupies the cell 46. The UAVs occupy initially the cells 54, 55, 37 and 38. In both figures, each UAV visits only adjacent cells and there is no overlap between paths. With regard to the number of cells visited by each UAV, a better fairness is achieved by the cluster formation.

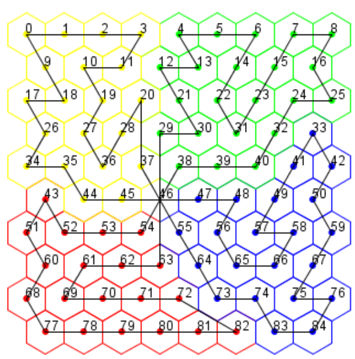

Fig. 3: Solution provided by the cluster formation method

Figure 5 illustrates the number of cells visited by each UAV in various configuration for both solutions. We observe that for small configurations (e.g. area size $\leq 250m*250m$), both solution provide very good results with a fairness index better than 0.98 as shown in Figure 6. But for large configurations (e.g. area size $\leq 400m * 400m$), the fairness index becomes poor (0.76) with the coalition game whereas it remains excellent for the cluster formation (1).

Similar results are obtained for the fairness index applied to the distance traveled by each UAV as shown in Figure 7.

As expected, the average distance traveled by a UAV increases when the size of the area to be monitored increases.

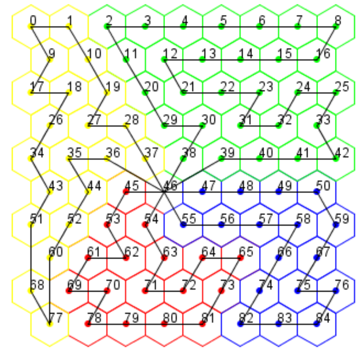

Fig. 4: Solution provided by the coalition formation method

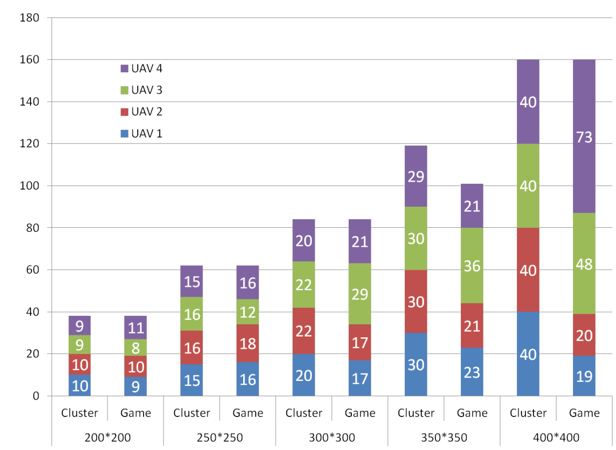

Fig. 5: Number of cells per coalitions for both methods

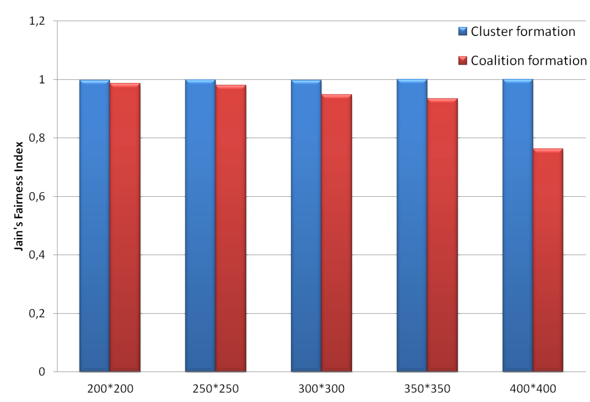

Fig. 6: Fairness in terms of number of nodes visited by each drone

We observe that the cluster formation solution provides smaller distances in all configurations tested as depicted in Figure 8. This can be explained by the fact that the cells visited by a UAV occupy a smaller zone with the cluster formation than with the coalition game.

Table I, gives the processing time of both solutions for various configurations. It clearly appears that the cluster formation is time efficient (with a processing time ≤ 0.047 seconds)

TABLE I: Processing time in seconds

Configuration	#Cells	Cluster formation	Coalition formation	# switches	# iterations
200*200	39	0.016	62.9	34	4
250*250	63	0.016	61.8	58	5
300*300	85	0.016	151	80	6
350*350	120	0.047	774	115	8
400*400	161	0.047	12300	156	9

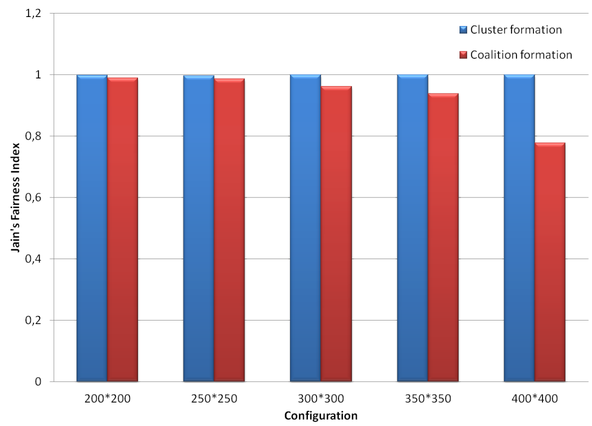

Fig. 7: Fairness in terms of distance traveled by each drone

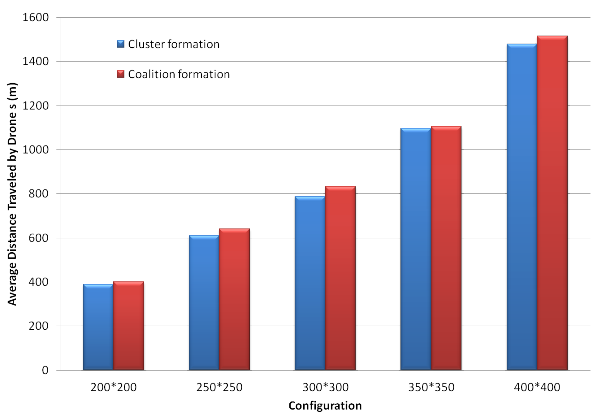

Fig. 8: Average distance traveled by drones

whereas the coalition game needs up to 12300 second for $400m * 400m$ configuration.

To explain this high processing time for the coalition game we compute the total number of switches and the number of iterations for each configuration. We observe that, in all configuration tested the number of swithes is equal to the number of cells minus 5. These 5 cells correspond to the cells occupied by the UAVs and the sink. Hence, each free cell switches only once. Similarly the number of iterations increases slightly with the number of cells. We can conclude that the principle of the coalition game scales. But the functions called to compute the utility function takes a lot of time: the computation of the centroid for each coalition requires to sort the set of cells to minimize the path visiting them. In this paper, we use the 2-Opt heuristic than is expensive when the number of cells increases. The processing time of the coalition game can be considerably improved by selecting a more efficient method to compute the payoff.

VII. Conclusion

In this paper, we show how to monitor a given area using Unmanned Aerial Vehicles (UAVs), also known as drones. In order to maximize the battery lifetime of each UAV, we need to minimize the distance traveled and the number of cells visited by each UAV. To achieve our goal, we proceed in two steps. In the first step, we proposed two solutions based either on cluster formation or on coalition game to assign to each UAV the cells to visit. In the second step, we optimize the path of each UAV visiting its cells based on an optimization heuristic. The two proposed solutions are compared in terms of the number of cells visited, distance traveled and processing time. Both solutions provide good results. However, the cluster formation solution performs better than the coalition game when the area size increases. This is due to an expensive computation of the payoff.

References

[1] N. Loizou and C. de la Vega, "The arrival of drones," 2014.

[2] A. Bürkle and S. Leuchter, "Development of micro uav swarms," in *Autonome Mobile Systeme 2009*, pp. 217–224, Springer, 2009.

[3] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *J. ACM*, vol. 7, pp. 326–329, Oct. 1960.

[4] N. Christofides, A. Mingozzi, and P. Toth, "Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations," *Mathematical Programming*, vol. 20, no. 1, pp. 255–282, 1981.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. 1990.

[6] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209 – 219, 2006.

[7] L. Tang, J. Liu, A. Rong, and Z. Yang, "A multiple traveling salesman problem model for hot rolling scheduling in shanghai baoshan iron and & steel complex," *European Journal of Operational Research*, vol. 124, no. 2, pp. 267 – 282, 2000.

[8] A. E. Carter and C. T. Ragsdale, "A new approach to solving the multiple traveling salesperson problem using genetic algorithms," *European Journal of Operational Research*, vol. 175, no. 1, pp. 246 – 257, 2006.

[9] Y. Wang and C.-H. Wu, "Robot-assisted sensor network deployment and data collection," in *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on*, pp. 467–472, June 2007.

[10] D. Bhadauria and V. Isler, "Data gathering tours for mobile robots," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 3868–3873, Oct 2009.

[11] I. Khoufi, M. Hadded, P. Minet, and A. Laouiti, "Optimized trajectories of multi-robot deploying wireless sensor nodes," in *2015 Wireless and Mobile Computing, Networking and Communications, WiMob 2015, Abu Dhabi, UAE, October 19-21, 2015*, pp. 1–6, 2015.

[12] G. A. Croes, "A method for solving traveling-salesman problems," *Operations Research*, vol. 6, 1958.

[13] W. Saad, H. Zu, T. Basar, D. Debbah, and A. Hjorungnesng, "Hedonic coalition formation for distributed task allocation among wireless agents," *IEEE Trans. on Mobile Computing*, vol. 10, 2011.