

# Multi-robot Task Planning Problem with Uncertainty in Game Theoretic Framework

Krzysztof Skrzypczyk and Martin Mellado

**Abstract.** An efficiency of an multi-robot systems depends on proper coordinating tasks of all robots. This paper presents a game theoretic approach to modelling and solving the pick-up and collection problem. The classical form of this problem is modified in order to introduce the aspect of an uncertainty related to an information about the workspace inside of which robots are intended to perform the task. The process of modelling the problem in game theoretic framework, as well as cooperative solution to the problem is discussed in this paper. Results of exemplary simulations are presented to prove the suitability of the approach presented.

**Keywords:** task planning, uncertainty, game theory, multi-robot.

## 1 Introduction

In multi-robot systems the primary, complex task is distributed between a number of robots that can perform some simple operations called further sub-tasks. Those systems benefit from the property that even complex process can be solved by a number of simple robotic units. Considering mobile robotics domain, many practical applications of such systems can be pointed out. That is enough to mention a problem of a transportation of large, and heavy parts by a number of smaller transporters, exploration task as well as pick up and collection problem. The results of application of

---

Krzysztof Skrzypczyk  
Silesian University of Technology, Institute of Automatic Control, Akademicka 16,  
44-100 Gliwice, Poland  
e-mail: krzysztof.skrzypczyk@polsl.pl

Martin Mellado  
Instituto de Automatica e Informatica Industrial, Universidad Politecnica de Valencia,  
Camino de Vera 14, 46022 Valencia, Espana  
e-mail: martin@ai2.upv.es

multi-robot systems are two fold. On the one hand we benefit from the innate, aforementioned features of these systems like scalability and their distributive nature. But on the other hand, there is a need of coordination of actions of particular units which is not a simple task. Considering distributed mobile robot systems, in many cases the problem of sharing the common workspace and resources appears. This implies that, without proper method of coordination even simple task of moving robots inside of an empty workspace can not be performed. The problem is getting more complex if the information provided to the system is uncertain. In spite of the problems related, advantages of these systems cause they have been intensively investigated. Therefore a series of different methods of solving the problem of coordination have been proposed suggested by different authors [1, 2, 7, 8, 9, 10, 16].

One of the tools which suites well to the discussed problem, is the Game Theory, which provides a smart framework for modelling and solving the problems that have conflicting nature. It is easy to notice that in multi-robot systems, mutual interactions between individual robots may result in conflict, which is often caused by the fact of sharing common resources. That is the reason many researches try to apply this framework to model different aspects of multi-robot systems [3, 4, 5, 11, 15]. In this paper an approach to model and solve the pick up and collection task in multi robot environment is presented. This problem is a popular benchmark in Multi-Agents Systems which allows to verify different control, coordination and task distribution strategies. Classical formulation of this problem, which has been investigated in many papers [12, 13, 14] does not take into account uncertainty of information about the workspace the team of robots is provided with [10]. To extend of the primary pick-up collection problem it is assumed that the number as well as location of the objects the team is intended to collect is uncertain. In this case the problem is getting more interesting, and hard to solve by off-line task planners. Such modification of the problem is the point of this paper. The rest of the paper is organized as follows. After introducing the basic control system framework a model of the workspace is described. Next part of the article define the tasks particular robots are able to perform. After this the problem statement is presented. Another part of the article describes the process of modelling the problem in the game theoretic framework. In the last section simulations that prove the appropriateness of presented approach are presented and discussed.

## ***1.1 The System Framework Overview***

In this section a brief overview of the system's framework is presented. Figure 1 illustrates conceptual architecture of the multi-robot control system. It is assumed that the system is the centralized one. That means that the control is determined by the module responsible for the task planning and then sent to the individual robotic units. The task planner computes the set of actions that are to be executed by each robot. That implies the system works in a synchronous way. The problem solver

cannot send the new set of actions to the group of robots, until the last robot reports completion of its task. Therefore the system should not be perceived as a distributed control system (multi-agent). The control is centralized and sent to the individual robots synchronously. On the other hand the information about the workspace is collected and then updated in a distributive way. The model of the workspace introduced to the system in the beginning of the process, is successively updated by individual robots. In the system described, the blackboard exchange mechanism is applied, what means that each robot has an access to some system variables, and is able to read them and overwrite them, while collecting new information.

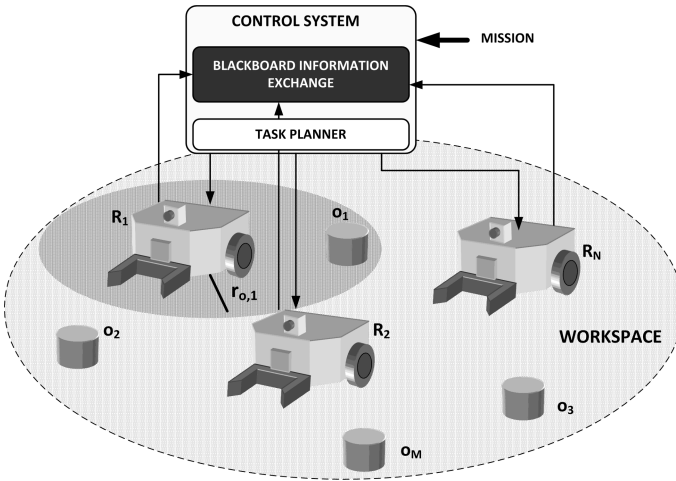


Fig. 1 Conceptual diagram of the system

## 2 Model of the Workspace

The pick-up and collection problem consists of collecting by the group of robots a number of objects scattered inside the workspace. The system is provided with information about the location of objects described in the Cartesian coordinates system fixed to the workspace. However, the information about the fact that there is the given object on the known position is not certain one. This uncertainty must be taken into account during the process of planning the task, what implies it must be included in the model of the workspace. Moreover each robotic unit is assumed to have some operational range, that describe its possibility of the movement during the single cycle of the process. This aspect of the limited mobility of robotic units must also be included in the model. Taking aforementioned aspects into account, the workspace is modeled as a weighted visibility graph, which determines

possible movements of individual units in the given state of the process. The graph is defined as  $W = (V, E)$  where  $V = \{v_1, v_2, \dots, v_M\}$  is the set of vertices of the graph, that represent  $M$  objects deployed inside of the workspace. The following features are related to the given vertex:

$$v_i = [x_{o,i}, y_{o,i}, P_{o,i}, r_{o,i}] \quad (1)$$

In (1) variables  $(x_{o,i}, y_{o,i})$  denote the position of  $i$ th object,  $P_{o,i}$  is the probability of the fact that  $i$ th object is located inside of the circle of the centre fixed in  $(x_{o,i}, y_{o,i})$  and the radius  $r_{o,i}$ . Since for considered problem both shape and orientation of objects is not important, each object is described only by position of geometrical centre. Second part of the model is the set of edges  $E$  that describes spatial relations between objects. As it was mentioned before, each robot is assumed to have a given operational range  $r_{R,i}$  for  $i = 1, 2, \dots, N$ , where denotes a number of robots. That means it can only reach an object that is closer than the range of the robot. Moreover, a robot cannot move directly to an object if there is another object on the way. The reason of such assumption is that such movement can disturb the searching or picking-up process proceeded by other robot. The set of objects that lay on the way between  $i$ th and  $j$ th object is denoted by:

$$V_{i,j}^{obst} = \{v_k, \dots, v_l, \dots, v_m\} \subset V \quad (2)$$

Thus the edges of the graph are defined:

$$E = \{v_i, v_j, w_{i,j} : v_i, v_j \in V, w_{i,j} \in \mathbf{R}\} \quad (3)$$

The weighting factor  $w_{i,j}$  in (3) is calculated using the formula:

$$w_{i,j} = \begin{cases} \|v_i - v_j\| & \text{if } \|v_i - v_j\| < r_{R,i} \wedge V_{i,j}^{obst} = \phi \\ \infty & \text{otherwise} \end{cases} \quad (4)$$

The weighting factor is equal to the distance between the object and the robot if this distance is smaller than operational range of the robot. Also must be assured that there is no any other object laying between the robot and the given object. Otherwise the weighting factor is equal to infinity. The model contains information about the workspace the system is provided with. The real state of the workspace is described by:

$$V^{real} = \{P_{r,1}, P_{r,2}, \dots, P_{r,M}\} \quad P_{r,i} = 0, 1 \quad (5)$$

Elements of this set describe the fact if really there is the given object inside of this workspace or not. In fact the set (5) denotes a true, initial state of the workspace and in general can differ from the description given by (1).

## 2.1 Robots and Their Actions

Within the workspace  $W$  operate  $N$  mobile robots. The group of robots is denoted as a set  $R = \{R_1, R_2, \dots, R_N\}$ . The state of  $i$ th robot is defined by its position and described by a vector:

$$X_i = [x_{R,i}, y_{R,i}] \quad (6)$$

where elements of the (6) denote the position of  $i$ th robot described in the coordinate frame of the workspace  $W$ . Since it is not important for considered problem, the orientation of the robot is neglected. In terms of the assumed model of the workspace as well as the process of task planning, the robot can be located only in finite number of places, which are defined by objects' location. Therefore, motion between these locations is assumed to be solved by a path planner avoiding any possible collision. Thus the position of the  $i$ th robot, in terms of the model, is defined

$$m_i = m, m = 1, 2, \dots, M \text{ for } \sqrt{(x_{R,i} - x_{o,m_i})^2 + (y_{R,i} - y_{o,m_i})^2} < r_{o,m_i} \quad (7)$$

Therefore the state of the team in the given discrete moment  $n$ ,  $n = 1, 2, \dots$  of the process is given by:

$$M(n) = \{m_i\}, i = 1, 2, \dots, N \quad (8)$$

Each robot depending on its state (7) is able to perform a given number of actions. The set of actions admissible in the given state  $m_i$  for  $i$ th robot is denoted by:

$$A_i^{m_i} = \{a_{i,1}, a_{i,2}, \dots, a_{i,K_i}\} \quad (9)$$

where  $K_i$  is a number of admissible actions of the  $i$ th robot in the state  $m_i$ . Each action the given robot can take is the one of the three following ones.

- *GoTo*( $R_i, v_j$ ) - is the action that consists in moving the  $i$ th robot close to the  $j$ th object. This action, in order to be admissible, must satisfy the precondition:  $w_{m_i,j} \neq \infty$ , whereas a result of applying this action is described by the postcondition:  $m_i = j$ ,  $P_{o,j} = P_{r,j}$
- *PickUp*( $R_i, v_j$ ) - is second action that can be taken by a given robot. It consists in picking up and collecting the object  $v_j$  by the robot  $R_i$ . This action can be performed if it is satisfied the following:  $m_i = j$ ,  $P_{o,j} \neq 0$   $v_{m_i} \notin V^{col}$  where  $V^{col}$  denotes the set of objects that have already been collected. Applying this action results in the postcondition:  $P_{o,j} = 0$ ;  $V^{col} = V^{col} \cup v_j$
- *Wait*( $R_i$ ) - this is third action the given robot can take. The result of applying this action is stopping the robot for one planning cycle. This action does not need any preconditions.

## 2.2 Problem Statement

Using the introduced notation the problem can be clearly stated as follows. For each discrete moment of time  $n$  which is the planning cycle, find a set of actions  $A_0(n) = \{a_{0i}\} i = 1, 2, \dots, N, n = 1, 2, \dots$  that if applied will result in satisfying the following:

$$V = V^{col} \quad (10)$$

It is required the condition (10) to be satisfied, after as small as possible number of planning cycles. Moreover the aim of the planning process is an uniform distribution of collected objects that reduce the amount of energy spent by the team. The (10) is a condition of terminating the collection process and means that all known objects are collected. It is important to stress that all the process of task planning is not the off-line one. Since there is an uncertainty of the information, the tasks are scheduled in the step by step mode, while the information is being updated by the robots performing the mission. That is the reason that the result of the task planning might not be optimal. It depends strongly on the level of uncertainty of information the system is provided with.

## 3 Modelling the Problem

The pick-up and collection problem is a typical example of a process when multiple robotic units must share common resources during execution the common mission. That implies mutual interactions of the units can lead to ineffective or even to inability of execution of the mission. In this section the process of modelling the pick-up and collection problem in game theoretic framework is described. The process of execution of the considered task can be perceived as a game between individual robotic units (players). Each of them is able to perform a set of the actions described in section 2.1. Depending on the given combination of actions taken by individuals and the current state of the process, the game results in some cost the team is trying to minimise (in case of team problem). Let us model considered process as a game related to the given discrete moment of the planning process:  $G(n) = \{N, A(n), I(n)\}$ , where where  $N$  denotes number of players (robots) taking part in the game,  $A(n)$  is an action space of the given  $n$ th step of the process:

$$A(n) = A_1^{m_1} \times A_2^{m_2} \times \dots \times A_N^{m_N} \quad (11)$$

where  $A_i^{m_i}$  is an action set of the  $i$ th robot in the state  $m_i$  defined by (9). Third component of the game is the cost function  $I$ . The game theory derives from the economical sciences and in this context the cost function denoted the amount of costs that the players had to pay for taking individual combination of strategies. In terms of robotics, the cost function has no physical interpretation, and the value the function returns is considered as a numerical quantity that is to be minimized. Designing the cost function is the key point in creating the game theoretic framework

for the considered problem. The value of the cost function, arguments of which are particular actions of the robotic units, reflects the quality of execution of the considered task. The form of this function must encapsulate all the features of multi robot execution of the mission. In discussed approach the following form of the cost function was applied:

$$I = \sum_{i=1}^N I_i(a_k), a_k \in A_k^{m_k} \quad k = 1, 2, \dots, N \quad (12)$$

Each component  $I_i$  is the cost function related to the  $i$ th robot. This component of the cost functions, consists of the two parts:

$$I_i(a_k) = I_{exp,i} + I_{rew,i} \quad (13)$$

The first one is related to profitability of the action applied. Since a presence of the object in the location is given with some probability, there is a need to estimate the costs of moving the robot to the given position. The higher probability of the presence of the object and the lower is the distance to it, the more profitable is taking the considered action. This component for the action  $a_k = GoTo(R_i, v_j)$  is defined by:

$$I_{exp,i}(a_k) = \begin{cases} \infty & \text{if } v_j = v_i, j, j = 1, \dots, N \\ \alpha \frac{P_{o,a_i}}{L_{i,v_i}} + \beta \frac{P_{o,v_i}^*}{L_{o,v_i}^*} + b & \text{otherwise} \end{cases}, \quad k = 1, 2, \dots, N \quad (14)$$

where  $P_{o,a_i}$  is the probability of existence of the object  $v_i$  selected by  $i$ th robot as a result of action  $a_k$ . The value  $L_{i,v_i}$  is the cost of moving  $i$ th robot to the given object  $a_i$ . This value is equal to the weighting factor  $w$ . The value of the other part of described component is related to proximity of the object  $v_i$ , to the closest uncollected object. Thus  $P_{o,v_i}^*$  and  $L_{o,v_i}^*$  denote the probability of existence of the object closest to  $v_i$  and distance between the closest, uncollected object and the object  $v_i$ . The role of coefficients  $\alpha, \beta$  as well as the bias  $b$  is adjusting the model. The value of this component for actions *PickUp*, and *Wait* is equal to 0.

The value given by second component in (13) can be described as a reward taken by individual player, for picking up the given object. This value depends on the state of the overall process, and can vary depending on the distribution of number of objects collected by individual robots. The aim is to provide uniform distribution and obtain the best possible value of the cost function. Therefore the value of this factor, defined for *PickUp* action is given by:

$$I_{rew,i}(a_k) = \begin{cases} -R & \text{if } \hat{\delta}_i > 0 \wedge m_{v0} = v_i \wedge \hat{L}_i^* < L_{v_k, v_{k0}} \\ R & \text{otherwise} \end{cases} \quad (15)$$

where  $R$  is some positive, real value. Therefore, this factor can be positive or negative according to whether the decision taken by  $i$ th robot is cooperative or noncooperative. The action is considered to be noncooperative one if the conditions pointed in (15) are satisfied. The first part of the condition determine if number of collected

objects by  $i$ th robot is not lowest one (including the object collected as a result of action  $a_k$ ). The value is determined from:

$$\hat{\delta}_i = \hat{N}_i - \min_{k=1, \dots, N, k \neq i} (\hat{N}_k) \quad (16)$$

where  $\hat{N}_k$ ,  $k = 1, \dots, N$  is the number of collected objects by  $k$ th robot, after taking the action  $a_k$ . Next conditions determine if the object  $v_i$  the  $i$ th robot is about to pick up is the closest uncollected object for other robot. To know if it is, it is checked if the distance  $\hat{L}_i^*$  to other closest object for  $i$ th robot is smaller than  $L_{v_k, v_{k0}}$ . Such form of the component stimulates the cooperative behaviour of the team. It allows to provide uniform distribution of number of collected elements as well as minimise the energy spent by a team. This term is equal to 0 for *GoTo* and *Wait* actions.

## 4 Solution

The problem stated in this paper is called in terms of game theory the team problem which consists in optimisation the common cost function. Moreover the considered problem is cooperative one with total exchange of information and successive synchronisation of task execution of individual robot. It is important to stress again that the task execution process is the iterative one with successive update of knowledge made by individual team mates. Therefore the solution of the single  $n$ th step of the process is given by:

$$A_0(n) = \{a_{10}, a_{20}, \dots, a_{N0}\} = \min_{a_1, \dots, a_N} I \quad (17)$$

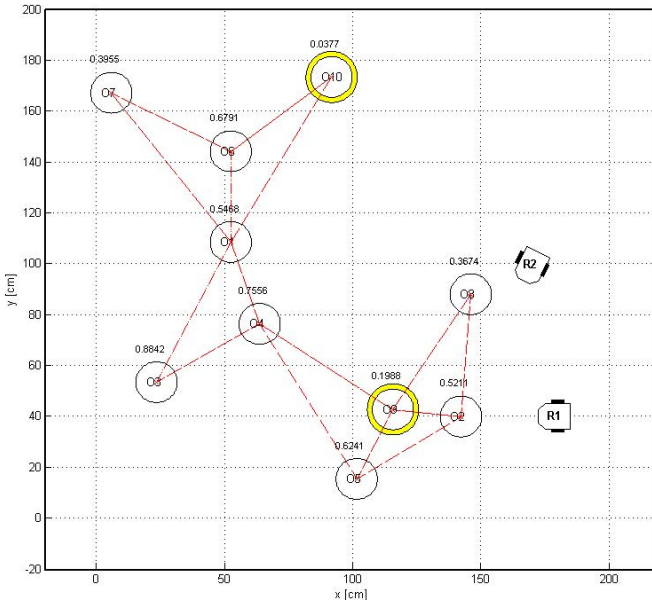
Applying the solution results in execution of the part of the primary problem and update the knowledge about the workspace. The solution (17) is applied until the stop condition is satisfied. The stop condition is defined by equality of the sets  $V = V^{col}$  what means that all the objects have been collected. The process also can stop when all the actions determined by (17) are *Wait* actions. This second situation may occur when there are objects described in model by very low probability factor. In this case if possible benefits from checking up the existance of the objects are lower than energy spent for this action the process may be stopped.

## 5 Simulation Results

In order to verify our approach the afore-described methodology was implemented in the MATLAB environment. A number of simulations have been performed to evaluate the proposed methodology of modelling and solving the pick up collection task. In this section only two, selected and relevant experiments are presented and discussed. In all presented simulations two robots  $\{R_1, R_2\}$  are intended to collect



a set  $\{v_1, v_2, \dots, v_M\}$  of objects where the number of objects is equal to 10. Inside the workspace, there are  $M$  objects distributed in a random way. The probabilities of existence are randomly fixed to the given objects. In all (excluding the first one) experiments 30% of objects are described by low probability factor. Among these objects there are objects that do not exist in the workspace. Thus difference between the real world and the model of the process is modeled. In first experiment the discussed method was verified using different scenario. This time ten objects were randomly displaced inside of the workspace. Existence of each object was described by probability factor (depicted by number outside the circle). Initial locations of robots are close to each other what implies some possible difficulties in task allocation. There are 2 objects described by low probability factor in the model, and which did not exist in reality (objects depicted with double circle). Modeled scenario is presented in fig. 2 while fig. 3 presents the solution of the problem, which is the sequence of robots actions. It is worth to notice that also in this scenario, the task was performed in a cooperative way. In stage 7 first robot leaves the object instead of picking it up and moves to 6th object. That allowed to obtain uniform distribution of collected objects between the robots.

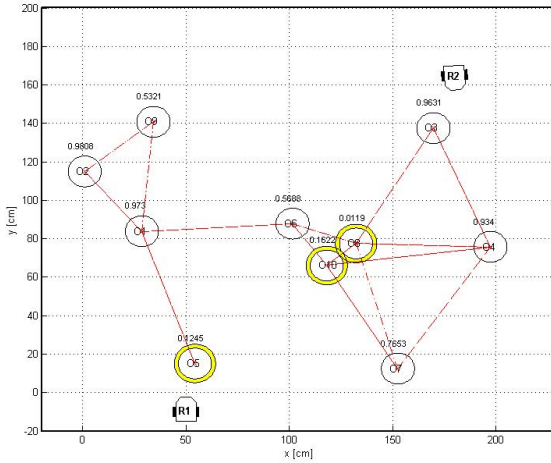


**Fig. 2** The layout of the objects and the model of the workspace used in first experiment

The goal of the next experiment is showing the functioning of the method in the case when the objects are split into two separated subgroups (fig.4). Also the robots start the work from two distant locations. In this experiment three objects were modelled as not existing ones, described by low probability factors. Analysing the

	ROBOT 1	ROBOT 2	
1	GoTo (R1, o2)	GoTo (R2, O8)	time ↓
2	PickUp (R1, o2)	PickUp (R2, o8)	
3	GoTo (R1, o5)	GoTo (R2, O9)	
4	PickUp (R1, o5)	GoTo (R2, o4)	
5	Wait (R1)	PickUp (R2, O4)	
6	GoTo (R1, o4)	GoTo (R2, o3)	
7	GoTo (R1, o1)	PickUp (R2, O3)	
8	GoTo (R1, o6)	GoTo (R2, O1)	
9	PickUp (R1, o6)	PickUp (R2, o1)	
10	GoTo (R1, o7)	GoTo (R2, o10)	
11	PickUp (R1, o7)	Wait (R2)	

**Fig. 3** Sequence of robot actions - result of execution of the task stated in first experiment



**Fig. 4** The layout of the objects and the model of the workspace used in second experiment

solution (fig. 5) it is easy to notice that the behaviour of the team is quite reasonable and intuitive in terms of human behaviour. Robot 1 after picking-up 9th object, waits and let the other robot collect the rest of objects. Thus energy saving aspect of the team behaviour can be noticed.

**Fig. 5** Sequence of robot actions - result of execution of the task stated in second experiment

	ROBOT 1	ROBOT 2
1	GoTo (R1 ,o5)	GoTo (R2 ,o3)
2	GoTo (R1 ,o1)	PickUp (R2 ,o3)
3	PickUp (R1 ,o1)	GoTo (R2 ,o4)
4	GoTo (R1 ,o2)	PickUp (R2 ,o4)
5	PickUp (R1 ,o2)	GoTo (R2 ,o7)
6	GoTo (R1 ,o9)	PickUp (R2 ,o7)
7	PickUp (R1 ,o9)	GoTo (R2 ,o10)
8	Wait (R1)	GoTo (R2 ,o6)
9	Wait (R1)	PickUp (R2 ,o6)
10		

## 6 Conclusion

In this paper an approach to model and solve the pick up and collection task in multi-robot environment was presented. The discussed problem is a popular benchmark in Multi-Agents Systems which allows to verify different control, coordination and task distribution strategies. The extended version of the problem, that consists in introducing uncertainty of the model was the key point of this article. It was assumed that the number as well as location of the objects the team is intended to collect is uncertain. A game theoretic approach to model and solve the aforementioned problem was applied. Because the knowledge of the system about environment is uncertain it is difficult to apply off-line planning. Instead of this an iterative step-by-step execution of the task was applied. Implications of such approach are two fold. On one hand it is possible to acquire knowledge about the environment and the real state of the process. On the other hand this approach can not guarantee the optimal execution of the task. Nevertheless a number of simulation, three examples of which are presented in the paper, confirm the appropriateness of the approach. Results of the simulations show that the method behaves in a rational (in terms of human behaviour) way and the obtained solutions are acceptable. Such good results are obtained of course in case of providing step by step synchronisation of the process. Future investigation will be focused on examining the problem when actions are not synchronised. In this case another uncertainty factor related to the effect of the actions taken by robots, should have to be considered.

## References

1. Alami, R., et al.: Toward human-aware robot task planning. In: Proc. of AAAI Spring Symposium, Stanford (USA), pp. 39–46 (2006)
2. Baiocchi, M., Marcugini, S., Milani, A.: Task Planning and Partial Order Planning: A Domain Transformation Approach. In: Steel, S. (ed.) ECP 1997. LNCS, vol. 1348. Springer, Heidelberg (1997)
3. Desouky, S.F., Schwartz, H.M.: Self-learning Fuzzy logic controllers for pursuit-evasion differential games. *Robotics and Autonomous Systems* (2010), doi:10.1016/j.robot.2010.09.006
4. Harmati, I., Skrzypczyk, K.: Robot team coordination for target tracking using fuzzy logic controller in game theoretic framework. *Robotics and Autonomous Systems* 57(1) (2009)
5. Kaminka, G.A., Eruslimchik, D., Kraus, S.: Adaptive Multi-Robot Coordination: A Game-Theoretic Perspective. In: Proc. of IEEE International Conference on Robotics and Automation, Anchorage, Alaska, USA (2002)
6. Kok, J.R., Spaan, M.T.J., Vlassis, N.: Non-communicative multi-robot coordination in dynamic environments. *Robotics and Autonomous Systems* 50(2-3), 99–114 (2005)
7. Klusch, M., Gerber, A.: Dynamic coalition formation among rational agents. *IEEE Intelligent Systems* 17(3), 42–47 (2002)
8. Kraus, S., Winkfeld, J., Zlotkin, G.: Multiagent negotiation under time constraints. *Artificial Intelligence* 75, 297–345 (1995)
9. Kraus, S.: Negotiation and cooperation in multiagent environments. *Artificial Intelligence* 94(1-2), 79–98 (1997)
10. Mataric, M., Sukhatme, G., Ostergaard, E.: Multi-Robot Task Allocation in Uncertain Environments. *Autonomous Robots* (14), 255–263 (2003)
11. Meng, Y.: Multi-Robot Searching using Game-Theory Based Approach. *International Journal of Advanced Robotic Systems* 5(4) (2008)
12. Jones, C., Mataric, M.: Adaptive Division of Labor in Large-Scale Minimalist Multi-Robot Systems. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, USA, pp. 1969–1974 (2003)
13. Sariel, S., Balch, T., Erdogan, N.: Incremental Multi-Robot Task Selection for Resource Constrained and Interrelated Tasks. In: Proc. of 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA (2007)
14. Schneider-Fontan, M., Mataric, M.J.: Territorial Multi-Robot Task Division. *IEEE Transactions on Robotics and Automation* 14(5), 815–822 (1998)
15. Song, M., Gu, G., Zhang, R., Wang, X.: A method of multi-robot formation with the least total cost. *International Journal of Information and System Science* 1(3-4), 364–371 (2005)
16. Cheng, X., Shen, J., Liu, H., Gu, G.-c.: Multi-robot Cooperation Based on Hierarchical Reinforcement Learning. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4489, pp. 90–97. Springer, Heidelberg (2007)