Application of Artificial Capital Market in Task Allocation in Multi-robot Foraging

Adel Akbarimajd
Electrical Engineering Department, Faculty of Engineering, University of Mohaghegh Ardabili,
Ardabil, Iran
E-mail: akbarimajd@uma.ac.ir

Ghader Simzan
Meshkin Shahr Branch, Islamic Azad University, Meshkin Shar, Iran
E-mail: simzan88@gmail.com

Received 4 April 2013

Accepted 21 December 2013

Abstract

Because of high speed, efficiency, robustness and flexibility of multi-agent systems, in recent years there has been an increasing interest in the art of these systems. Artificial market mechanisms are one of the well-known negotiation multi-agent protocols in multi-agent systems. In this paper artificial capital market as a new variant of market mechanism is introduced and employed in a multi-robot foraging problem. In this artificial capital market, the robots are going to benefit via investment on some assets, defined as doing foraging task. Each investment has a cost and an outcome. Limited initial capital of the investors constrains their investments. A negotiation protocol is proposed for decision making of the agents. Qualitative analysis reveals speed of convergence, near optimal solutions and robustness of the algorithm. Numerical analysis shows advantages of the proposed method over two previously developed heuristics in terms of four performance criteria.

Keywords: Task Allocation, Foraging Robots, Distributed Artificial Intelligence; Multi-agent Systems; Capital Market Mechanism

1. Introduction

Swarm robotics is a state of the art in robotics research. Large group of robots, when working with together, potentially can emerge complicated behaviors. The robots need cooperation, coordination and communication protocols to act as a unit.

Among well known swarm behaviors one can refer to multi-robot foraging. Foraging in mobile robots refers to task where the robots aim to find objects in a two-dimensional environment and collect them to specific goal points[1]. Different assumptions could be made in foraging task in terms of communication level among robots, motion pattern of objects, actuation and sensory constraints of robots, etc. Different assumptions impose different levels of complexity in foraging task.

Foraging task is an excellent test-bed in multi-robot systems. The reason is that foraging comprises some interesting and sophisticated sub-problems such as energy efficiency[2-6], path and motion planning[7-12], coordination[10, 13-16], communication[17-25], optimization[3, 7-9, 26-31] and task allocation[32-43]. Although in most cases these sub-problems are not completely independent, there have been a focus on a particular sub-problem in different researches in swarm robotics field.

One of the most interesting sub-problems in multi-robot foraging is task allocation. Task allocation refers to assigning available tasks to robots, such that a specific criterion is satisfied or optimized subject to some constraints. Some custom criteria are minimum time, maximum traveled distance and minimum consumed energy or their combinations. The constraints

Co-published by Atlantis Press and Taylor & Francis

may denote robot limitations in terms of power, velocity, force etc or they may correspond to environmental limitations such as obstacles, friction, etc.

There are two general approaches to solve task allocation problem including centralized and distributed approaches. At this point, employing of distributed paradigm is necessary for tasks those cannot be completed by a single robot[44]. Nevertheless, even when the task can be completed by a single robot, distributed approach is still beneficial as it may boost up efficiency, speed, flexibility and robustness of the system[45-46]. Distributed solutions of the problem as well as parallel computations would improve the efficiency and flexibility where the flexibility is related to novel and different solutions[46-49]. Distributed mechanisms facilitate repairing of failures hence a fault tolerant or a robust mechanism is achieved[50].

Despite having some advantages, distributed task allocation encounters some challenges. A main challenge is task decomposition where a total mission is reduced into some independent sub-tasks and each robot undertakes a subtask. In fact, robots perform their own jobs while completing whole mission of the mechanism. After task decomposition, a distributed decision making protocol is required in order to efficient assignment of available tasks. Therefore, a well-organized task allocation algorithm requires an appropriate model with well defined subtasks and efficiently developed assignment procedure.

In this paper we propose artificial capital market as a novel scheme for modeling of multi-agent systems. We also develop a negotiation protocol via which the agents can reach an agreement in the capital market. Then we will use artificial capital market as a task allocation mechanism in a multi-robot foraging system.

The rest of the paper is organized as the following. The next section includes related works. In Section 3 the proposed artificial capital market is described and in the forth section an appropriate negotiation algorithm for our capital market is devised. In Section 5 a complicated multi-robot foraging environment is considered and application of artificial capital market in task allocation in the environment is devised. The sixth section includes simulation results. Simulations are organized in two parts including qualitative and numerical studies. Conclusions are provided in the last section.

2. Related works

2.1. Local heuristics in task allocation

Local heuristics are a class of methods those have been successfully employed in distributed task allocation problems[51-52]. According to Ref 34, four local heuristics have been used in the literature:

- ***Closest Task First (CT):*** In this heuristic each robot selects an available task that is closest to it.
- ***Most Starved Task First (MST):*** In this heuristic, in order to balance the task load across the environment, the robots are more likely to perform the tasks those have the least number of robots in their neighborhood and necessitate more robots to complete them.
- ***Most Starved, Most Complete Task First (MSMCT):*** This is an extension of MST where the number of robots, required to complete a task, is also taken in account. Besides the tasks in the neighborhood, tasks those are near to be completed are given higher priority.
- ***Most Proximal Task First (MPT):*** In previous heuristics, each robot only considers status of available tasks. However in MPT a robot finds out which robots are closer to the task than the robot itself. Then it selects the task with the least number of robots closer to the task than itself.

2.2. Market mechanisms

Among distributed task allocation procedures, market inspired mechanisms have attracted much attention. Since firstly introducing this approach in Ref 53, different related researches have been reported. Most of them, in a general view, can be fall into three major categories:

- ***Auction mechanisms:*** Auction based multi-agent systems are one of the frequent market-based task allocation protocols. In this protocol, an auction is conducted where tasks (as goods) are sold to robots (as buyers). In an auction based system, an auctioneer agent supplies tasks to agents who can potentially perform the tasks. Each agent bids over available tasks and if it wins the auction, it performs the tasks.

  TraderBots is a well known example[47] that improves robustness and efficiency of multi-agent system by taking the advantage of efficiency and flexibility of the market. Jones et al exploited this protocol to obtain thriving solutions to form pickup teams in multi-robot systems[54]. In 2003 combinational auctions was developed to decrease computational cost of auction mechanisms; afterwards it was employed in robot exploration[55]. Auction based protocols have also been used for assigning dynamic tasks to multiple agents in formation control scenario[38] and for multi-robot dynamic coordination to visited locations in a map[56]. In Ref 57 a behavior based mechanism was proposed in order to implement a multi-robot cooperation system. This architecture is also exploited for development of auction based task assignment strategies in multi-robot systems applications. Zhang et al introduced a stochastic clustering auction for centralized and distributed task allocation in multi-agent teams[58] and employed it to improve the global cost of the task allocations obtained with fast greedy algorithms[59]. In Ref 60 an optimized decision mechanism based on multi-attribute auction was proposed for procuring of some divisible goods. Auction based mechanisms have also been used for inspection of graphic-like maps by multi-robot systems[61].

- ***Contract nets:*** The idea of using contract nets in multi agent systems was firstly introduced by Smith in 1980[61]. In this protocol manager agents (robots) announce an available task to other agents those operate as contractors. The contractors bid over the tasks if they are interested to do it. Then if the manager accepts the bid (according to a specific eligibility criterion) a contract is formed and the task is performed by the contractor agent.

As an earliest application, Parunak was employed contract nets for implementation of a control system in a discrete manufacturing environment[63]. In Ref 64 a task allocation graph was proposed to analyze different contract types including single task contracts, cluster contracts, swaps and multi-agent contracts. Then it was shown that no contract type can avoid local optima. In Ref 65 it was shown that the behavior of the protocol depends on system size and agent load. It was also illustrated that under heavy loads, the algorithm usually should be performed frequently. Contract nets were employed to develop a distributed task allocation scheme in multi-UAV robots in Ref 66. Hsieh has noticed that with the lack of a model in contract net protocol, it is difficult to assure feasibility of contracts. Then he presented a model to ease the bid evaluation to handle tasks with complex process structure[67]. In Ref 68 contract nets were used for coordination of agents in multi-agent systems. In Ref 69 the authors were proposed an adaptive efficient job scheduling service model on grids using a market like service level agreement (SLA) negotiation protocol based on contract nets. Lili and Huizhen proposed a threshold-limited load balance strategy for awarding in contract net models and established a mathematical model of negotiation process based on the improved contract net[70]. In Ref 71 a novel approach based on maximum dynamic integrative trust was introduced for contractor selection in contract nets in an agent based simulation environment.

- ***Market price equilibrium:*** In this approach to task allocation, some producer agents are defined in the environment those produce tasks in the market. On the other hand some consumer agents are defined those are interested to buy tasks. An adjustor agent tunes the prices in the market so that the market reaches to general equilibrium. In 1993, Wellman developed a market-oriented programming approach to distributed problem solving. He allocated tasks and recourses to a set of computational agents by obtaining equilibrium of an artificial market[72]. He then studied the concept of conjectural equilibrium in multi-agent learning systems[73]. Kaihara modeled problem of supply chain in a dynamic environment as a virtual market[74]. A context-aware coordinated web service multi-agent mechanism based on artificial market was proposed in Ref 75.

In all above protocols, there are some agents who are responsible for processing information and making decisions. Auctioneers in auction mechanism, managers in contracts net and adjustors in artificial price market are such agents. Therefore, none of these protocols are completely distributed and they always involve some central processing.

Artificial capital market, that we are firstly introduced in Ref 76, is a new paradigm in market based mechanisms. Artificial capital market is a completely distributed protocol as well as generating near optimal results in many cases.

3. Artificial Capital Market

In this section we describe essentials of artificial capital market as a mechanism for modeling and decision making purposes in multi-agent systems. In artificial capital market, agents are considered as investors those decide about some bundle of assets they are going to invest on[76]. Their decision is based on the expected profit of different bundles.

The components of artificial capital market are defined as the following:

- Investors: Agents in multi-agent systems are considered as investors.
- Assets: In multi-agent systems total task of the system is decomposed into some small subtasks. In the artificial capital market each subtask is considered as an asset.
- Cost of assets: Doing a specific task has a cost for agent. This cost can be different in different problems. For example in a multi-robot object manipulation problem the cost could be the consumed energy by robots; in transportation problem the cost could be the covered distance by vehicles and in software agents the cost could be related to computational complexities.
- Asset Bundles: Each agent can decide to participate in doing more than one subtask according to its capabilities. An asset bundle is composed of assets (subtasks) that the investor (agent) is going to invest on them.
- Bundle cost: Cost of a bundle is aggregate costs of assets included in the bundle.
- Initial capital: It is assumed that each investor have an initial capital I_i. The initial capital could be battery charge, computational capacity etc depending on type of agents.
- Outcomes: Investment on a bundle or equivalently doing corresponding subtask will have an outcome for the agent. Outcomes of different subtasks could be different. The agent should take in account this outcome in its decision making.
- Payoff: Payoff of agent i is defined as difference of its outcome and its costs:

$$P_i^j = \begin{cases} I_i + O_i^j - C_i^j & \mathit{if}\ \text{the agent}\, i\ \text{invests on bundle}\, j \\ 0 & \mathit{if}\ \text{the agent}\, i\ \text{does not invest on bundle}\, j \end{cases} \quad (1)$$

where O_i^j, P_i^j and C_i^j are payoff, outcome and cost values respectively.

4. Negotiation algorithm

In the artificial capital market, agents should decide about the bundles they are going to invest. In a multi-agent system it is desired that the decisions are made in a distributed manner and no central processing exists. It is assumed that each agent can obtain its local information including the costs and profits of assets available for itself. The only information that each agent gets from the other agents is their decisions. Each agent simply calculates the payoff of available bundles. If it has enough initial capital and investment on bundles are profitable, the agent decides to invest on bundles. Here two different cases exist:

- More than one agent can invest on a single bundle: In this case the profit of the bundle should be shared among the investors.
- Only one agent can invest on a single bundle: In this case if an agent decides to invest on a specific bundle, other agents have to give up investing on that bundle.

In both above cases, decision of each investor is highly dependent on decisions of other agents. Therefore a negotiation mechanism is required to support the society to come to an agreement about the decision of agents. Despite self-interested investors in the capital market, the negotiation mechanism should lead the agents to a decision which is superior for the society. This factor is the most important feature of the negotiation algorithm.

4.1. Decision making in static market

In static markets, the agents make their decisions once in the beginning and they do not have to change their decisions. In order to explain how the agents make decisions, a binary string $D^{k,j} = b_1^k b_2^k \ldots b_R^k$ is exploited to encode decisions of society about bundle j in iteration k. Bit b_i^k stands for decision of i^{th} agent in iteration k:

$$b_i^k = \begin{cases} 1 & \text{if agent } i \text{ decides to invest on bundle } j \\ 0 & \text{if agent } i \text{ decides not to invest on bundle } j \end{cases} \quad (2)$$

For example in a six agent system (A=6) $D^{7,5}$=001011 means that in iteration k=7 agents 3, 5, 6 have decided to invest on bundle 5. Decision string $D^{k,j}$ is a common

```
STATIC_MARKET
INITIATION: Select a random decision string $D^{0,j}$ as an initial string
For j=1:J
    WHILE $D^{k,j}$ has not been converged to a specific decision DO
      k=k+1
        FOR investors i=1:A DO
            Calculate $P_i^j$ using (1)
            IF $P_i^j < 0$ or $I_i - C_i < 0$
            THEN $b_i^k = 0$
        END OF FOR
    END OF WHILE
END OF FOR
```

Fig. 1. Negotiation algorithm for decision making in static markets based on artificial capital market

```
DYNAMIC_MARKET
WHILE Final Goal of the society is not achieved
    RUN STATIC_MARKET
    Execute selected tasks for $t_{fr}$ time duration
END OF WHILE
```

Fig. 2. Decision making algorithm in dynamic markets

knowledge for all agents. Based on this encoding, our instance negotiation algorithm starts with a random decision string $D^{0,j}$ for bundles j=1,2,3,…J where J is number of available bundles. Subsequently, the first investor calculates the costs of its currently available bundles. In the case that it has enough capital (i.e. $I_1 - C_1 > 0$) the investor calculates its payoff and selects the bundles it is going to invest on. The payoffs can be considered as local cost functions that the agents are going to minimize them. If the investment does not cover the costs, the investor suspends the investment. It is noteworthy to repeat that the investor performs all of calculations based on the current decision string as a common knowledge. If the investment on bundle j is profitable the investor accepts it and sets b_i^0 to one. If the investment on bundle j is not profitable the investor declines it and sets b_i^0 to zero. The next agent performs the same calculations using new decision string. Agents 1,…, A repeat this procedure in a loop until a decision string is converged to a fixed string. In fact this negotiation is a way to share local information of the agents. Pseudo code `STATIC_MARKET` shown in Figure 1 illustrates proposed decision making algorithm.

4.2. Decision making in dynamic market

In dynamic markets, agents have to change their decisions in accordance to environment changes. The investors simply employ `STATIC_MARKET` as basic subroutine to make decisions and choose appropriate tasks. They afterwards execute the tasks for t_{fr} time duration. At the end of period t_{fr}, they make new decisions using `STATIC_MARKET`. The procedure is repeated in a loop until the final goal is achieved. It is noteworthy to mention that the value of parameter t_{fr} depends on the rate of changes of the market. Fast changes in the market necessitate large values of t_{fr}. Figure 2 illustrates the proposed algorithm for dynamic markets.

5. Task allocation in multi-robot foraging system

Task allocation in multi-robot multi-object foraging problem with dynamic objects is a complicated problem that can be a good test bed for multi-agent problems. Different constraint assumptions can bring about different levels of complexity in foraging systems. Our case is one of the most complicated cases where R robots are located in an environment in different locations $\mathbf{P}_{Ri}=(X_{Ri}, Y_{Ri})$ as it is shown in Figure 3. The robots should collect the objects from positions

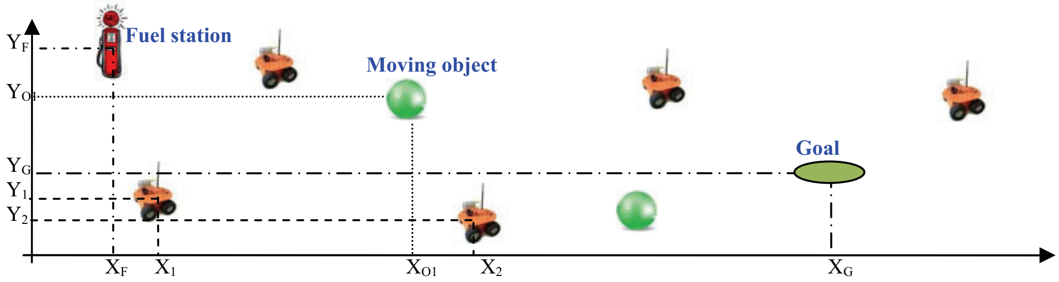


Fig. 3. Environment of multi-robot foraging problem

$\mathbf{P}_{Oj}=(X_{Oj},Y_{Oj})$ to a single goal point $\mathbf{P}_G(X_G,Y_G)$. The objects can be static or dynamic and may be carried by a single robot or by a team of robots. Robot *i* initially has limited fuel F_i. The fuel consumption rate of the robots while travelling with no load is E_f unit per meter. If a single robot carries a single object it would consume E_c unit of fuel per meter and if *n* robots cooperate in carrying an object the rate of fuel consumption will be decreased to E_c/n unit per meter for each robot. A fuel station is located at $\mathbf{P}_F=(X_F,Y_F)$. Only agents (robots) those have contributed in collecting the objects are allowed to fuel after completing the collection task. Each robot knows the positions of the goal and fuel station and has a local database including its own position. Database of each robot does not include positions of the other robots. The components of artificial capital market for this problem are defined as:

Investors: Robots

Assets: Covered distances by the robots, traveling with or without carrying the objects.

Cost of assets: Fuel consumption in a distance which depends on three parameters: covered distance, carrying or not carrying the objects and number of robots contributing in carrying a single object.

Asset Bundles: Contributing in transportation of an object. Each bundle includes three probable assets: finding the objects, taking them to the collection point and going back to the fuel

Table 1. Initial positions of robots. $\mathrm{P_i}=(X_i,Y_i)$ is position of i^{th} robot.

$\mathbf{P}_1$	$\mathbf{P}_2$	$\mathbf{P}_3$	$\mathbf{P}_4$	$\mathbf{P}_5$	$\mathbf{P}_6$	$\mathbf{P}_7$
(30,140)	(50,40)	(70,100)	(100,150)	(120,70)	(180,120)	(200,20)

Table 2. Positions of goal, object and fuel station in two problem cases studied in simulations.

CASE	Goal position $\mathbf{P}_G=(X_G,Y_G)$	Object position $\mathbf{P}_O=(X_O,Y_O)$	Fuel station position $\mathbf{P}_F=(X_F,Y_F)$
1	(10,70)	(110, 90)	(190,70)
2	(90,10)	(60,160)	(160,10)

Table 3. Progress of the proposed algorithm for first set of fuels in problem case 1 in three different runs.

Iterations	**Run 1** Converged in iteration **3**	**Run 2** Converged in iteration **1**	**Run 3** Converged in iteration **1**
$\boldsymbol{D^0}$	0100111	1011010	0111010
$\boldsymbol{D^1}$	0000100	0011100	0011100
$\boldsymbol{D^2}$	1010100	0011100	0011100
$\boldsymbol{D^3}$	0011100	0011100	0011100
$\boldsymbol{D^4}$	0011100	0011100	0011100
Final decision of the algorithm: 0011100		Fuel Consumption: 703.0478	
Optimal decision for the society: 0010100		Fuel Consumption: 653.4456	

Table 4. Progress of the proposed algorithm for first set of fuels in problem case 2 in three different runs.

Iterations	**Run 1** Converged in iteration **1**	**Run 2** Converged in iteration **1**	**Run 3** Converged in iteration **1**
$\boldsymbol{D^0}$	1010101	1111000	1010100
$\boldsymbol{D^1}$	1011000	1011000	1011000
$\boldsymbol{D^2}$	1011000	1011000	1011000
$\boldsymbol{D^3}$	1011000	1011000	1011000
$\boldsymbol{D^4}$	1011000	1011000	1011000
Final decision of the algorithm: 1011000		Fuel Consumption: 665.82	
Optimal decision for the society: 1011000		Fuel Consumption: 665.82	

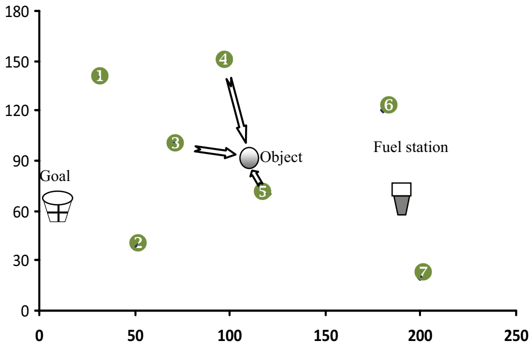


Fig. 4. Final solution of problem case 1 (first set of fuels)

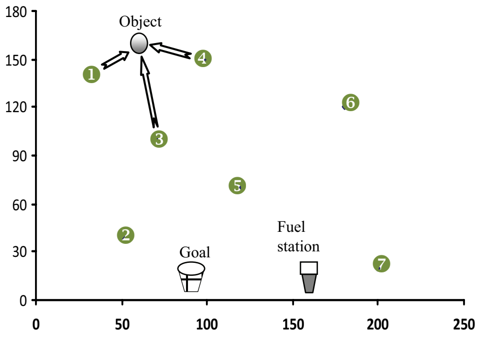


Fig. 5. Final solution of problem case 2 (first set of fuels)

station.

Initial capital: Initial fuel.

Outcomes: Amount of fuel in the station is limited that would be shared among robots those have participated.

Payoffs: Difference between outcomes of an agent and its costs.

6. Simulations

We have organized simulations in two subsections. In the subsection 6.1 we focus on the proposed method itself to show the progress of the algorithm and study its quality and suitability for the foraging system. Then in subsection 6.2, from a statistical viewpoint, we compare the efficiency of the proposed algorithm with two earlier methods.

6.1. Qualitative analysis

6.1.1. *Static market*

In order to verify efficiency and precision of the proposed method, we considered an environment with only one stationary object. The environment is a *200m×160m* area with 7 robots those initial locations are shown in Table 1. The rate of fuel consumptions with and without carrying the object are assumed E_f=0.7 and E_c=7 respectively. It is assumed that initial fuel of robots is 750 *(F_i=750 i=1,...,R)*. Available fuel at station is set to 800. We applied the proposed algorithm for two problem cases with different values for locations

Table 5. Progress of the proposed algorithm for second set of fuels in problem case 1 in three different ru

Iterations	**Run 1** Converged in iteration **3**	**Run 2** Converged in iteration **3**	**Run 3** Converged in iteration **3**
D^0	1010011	0110000	1111101
D^1	0011100	1111100	0011100
D^2	0011010	0011010	0011010
D^3	0011110	0011110	0011110
D^4	0011110	0011110	0011110
Final decision of the algorithm: 0011110		Fuel Consumption: 822.8696	
Optimal decision for the society: 0011100		Fuel Consumption: 703.0478	

Table 6. Progress of the proposed algorithm for second set of fuels in problem case 2 in three different runs.

Iterations	**Run 1** Converged in iteration **2**	**Run 2** Converged in iteration **3**	**Run 3** Converged in iteration **2**
D^0	0000100	1100001	0110000
D^1	1111100	1111000	1111100
D^2	1011100	1111100	1011100
D^3	1011100	1011100	1011100
D^4	1011100	1011100	1011100
Final decision of the algorithm: 1011100		Fuel Consumption: 685.8721	
Optimal decision for the society: 1011000		Fuel Consumption: 665.8200	

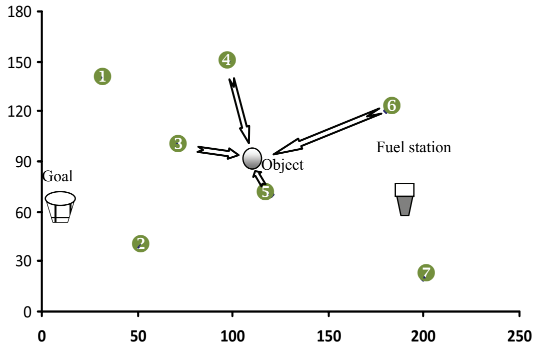


Fig. 6. Final solution of problem case 1 (second set of fuels)

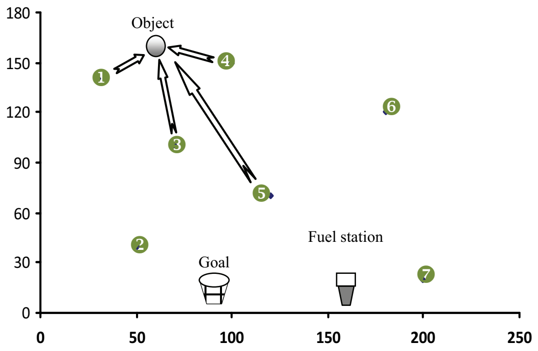


Fig. 7. Final solution of problem case 2 (second set of fuels)

of the object, goal point and fuel station illustrated in Table 2.

Progress of the proposed algorithm for different choices of initial decision string D^0 are shown in Tables 3 and 4. In both cases the algorithm is converged in less than 4 iterations. In problem case 1, robots 3, 4, 5 are decided to contribute in carrying the object (see Figure 4) and in problem case 2, robots 1, 3, 4 are decided to contribute in carrying the object (see Figure 5). These solutions do not depend on initial decision string D^0. In each case fuel consumption of the society corresponding to final decision is compared with optimal solution. The optimal solutions are obtained by a brute force approach where all solutions are calculated and the best one is selected. In case 1, fuel consumption corresponding to the final decision of the algorithm 703.0478 is about 7% more than that of optimal solution (653.4456). In case 2, fuel consumption corresponding to the final decision of the algorithm 665.82 is equal to that of optimal solution.

Subsequently, initial values of robot fuels are set to (F_1= F_2= F_6= F_7=800, F_3= F_4=F_5=300) and available fuel at fuel station is changed to S=900. For above problem cases 1 and 2, the progress of the algorithm and corresponding final decisions are shown in Tables 5 and 6. It can be concluded that the convergence is still fast and different initial fuels have resulted in different final decisions. In fact, different payoff of investors has caused different decisions. By changing robot fuels in the problem case 1, robot 6 has become interested to participate in transportation task. As a result, robots 3, 4, 5, 6 are included in the final decision (see Figure 6). In the same way in problem case 2, robot 5 has become interested to participate in transportation task and robots 1, 3, 4, 5 are included in the final decision (see Figure 7). For more details about static market simulations see Ref. 76 which is a preliminary work of this paper.

6.1.2. *Dynamic market: general problem*

We considered a general problem where the robots try to collect some moving objects. Dimension of the environment is considered to be *200m*×*200m* area with 10 robots. Initial locations of the robots are shown in Table 7. Values of E_f and E_c are set to 0.7 and 7 respectively. It is assumed that the robots have the same amount of initial fuel (F_i=750, i=1,...,R) and available fuel at the station is set to S=800. Positions of the fuel station and goal are $\mathbf{P}_F$=(160,140) and $\mathbf{P}_G$=(20,70) (see Table 8).

As it is explained in section 5.2, we employ decision making procedure of static market as core of decision making in dynamic market. Every t_{fr} time steps, negotiation is set up among the robots and a task allocation plan is generated according to current positions and fuels of the robots and positions of the objects. Once the tasks are allocated, the robots move toward the specified objects for t_{fr} time steps. Then a new allocation plan is generated. This procedure is continued until the robots reach the objects and collect them in collection points.

Single object simulations: In this part we provide single moving object simulations. Initial position of the object is $\mathbf{P}_O$=(100,100). In the first test it is assumed that the object randomly takes one of the eight available directions (N, S, W, E, NW, NE, SW and SE) to move one unit at each time step. In these simulations t_{fr} is set to 1 time steps. As new allocation plan is generated at each time step, simulations are relatively slow and the generated paths are near optimal. Five snapshots of simulation in MATLAB corresponding to time steps 25,

Table 7. Initial positions of robots. $P_i=(X_i,Y_i)$ is position of i^{th} robot.

$\mathbf{P}_1$	$\mathbf{P}_2$	$\mathbf{P}_3$	$\mathbf{P}_4$	$\mathbf{P}_5$	$\mathbf{P}_6$	$\mathbf{P}_7$	$\mathbf{P}_8$	$\mathbf{P}_9$	$\mathbf{P}_{10}$
(10,60)	(10,140)	(20,20)	(20,180)	(50,10)	(140,190)	(150,20)	(180,30)	(180,180)	(190,80)

Table 8. Initial position of the moving object and positions of goal and fuel station.

Goal position $\mathbf{P}_G=(X_G,Y_G)$	Object position $\mathbf{P}_O=(X_O,Y_O)$	Fuel station position $\mathbf{P}_F=(X_F,Y_F)$
(20,70)	(100,100)	(160,140)

50, 75, 100 and 125 are shown in Figure 8. It can be seen that the robots R7, R9, R10 have approached the object and the robots R7 and R10 have reached it before step 125. R4, R5, R6, and R8 have moved toward the object, however they have given up after some steps. R1, R3 and R2 have not moved toward the object.

In the second test, the object is assumed to take a direction and move in a straight line. Five snapshots of simulation corresponding to time steps 25, 50, 75, 100 and 125 are shown in Figure 9. It can be observed that the robots R2, R6, and R9 have caught the object. R2 had not been interested to contribute in the beginning (until time step 60) but by moving the object toward R2, it has become beneficial for R2 to move toward the object.

Two object simulations: Two moving object simulations are provided in this section. Initial position of the objects are $\mathbf{P}_{O1}$=(80,80) and $\mathbf{P}_{O2}$=(120,120). Similar to single-object simulations, two tests were performed; one with random motions of the objects and the other one with motions of the objects in straight lines. Figure 10 shows four snapshots of the first test. It can be seen that the robots R1, R2 and R3 have approached the object 1 and the robots R8 and R9 have approached the object 2. Figure 11 shows four snapshots of the second test. Yet again R1, R2 and R3 have caught the object 1 and the robots R8 and R9 have caught the object 2. It is interesting that the robots R1 and R3 have firstly caught the object 1 at time step 51 and the robot R3 has joined them later. The reason is that the investment has remained beneficial for R2 after time step 51.

6.2. Numerical analysis

Consider an environment with R robots and one static object where robots are homogenous and their initial fuels are equal. We denote area of the environment by A_e. A robot, With its initial fuel, can achieve the object and carry in to goal if it is located in a specific neighborhood of the object. We call this region as achievable region and denote its area by A_a. We also call the object as achievable object. A schematic diagram of such an environment is shown in Figure 12.a.

6.2.1. *Expected average values*

Assuming that all robots those are located inside the achievable area contribute in carrying the object we are going to find expected averages of number of robots contributing in foraging, steps taken by robots and fuel consumption by robots.

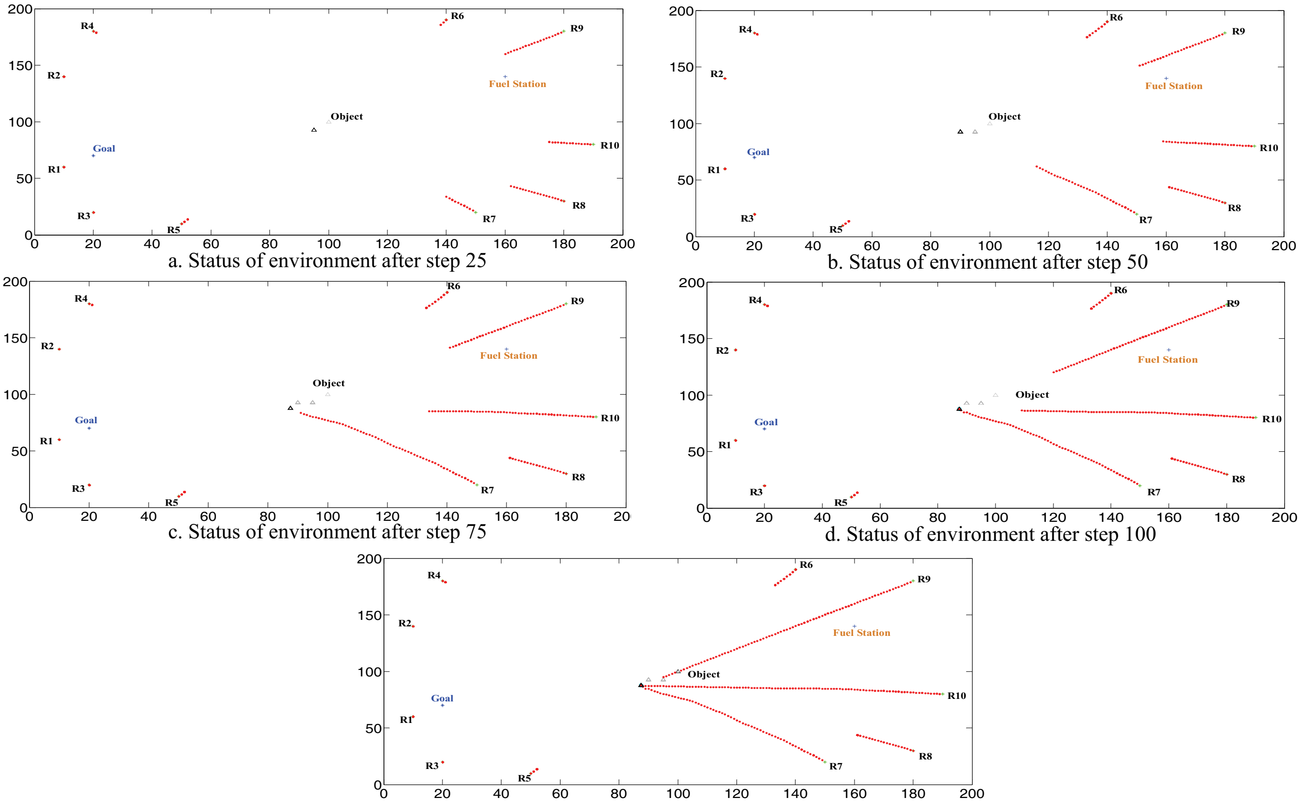
It is obvious that the probability of a single robot (R=1) being inside achievable area (A_a) is:

$$\mathrm{P} = \mathrm{P}(A_a) = \frac{A_a}{A_e} \tag{3}$$

The probability of r robots ($0 \le r \le R$) being inside achievable area is:

$$\mathrm{P}(r) = \binom{R}{r} \mathrm{P}^r (1-\mathrm{P})^{R-r} \tag{4}$$

In fact, $\mathrm{P}(r)$ is the probability of contributing r robots in carrying the object. Then, expected average number of robots those contribute in carrying the object is obtained by:



a. Status of environment after step 25

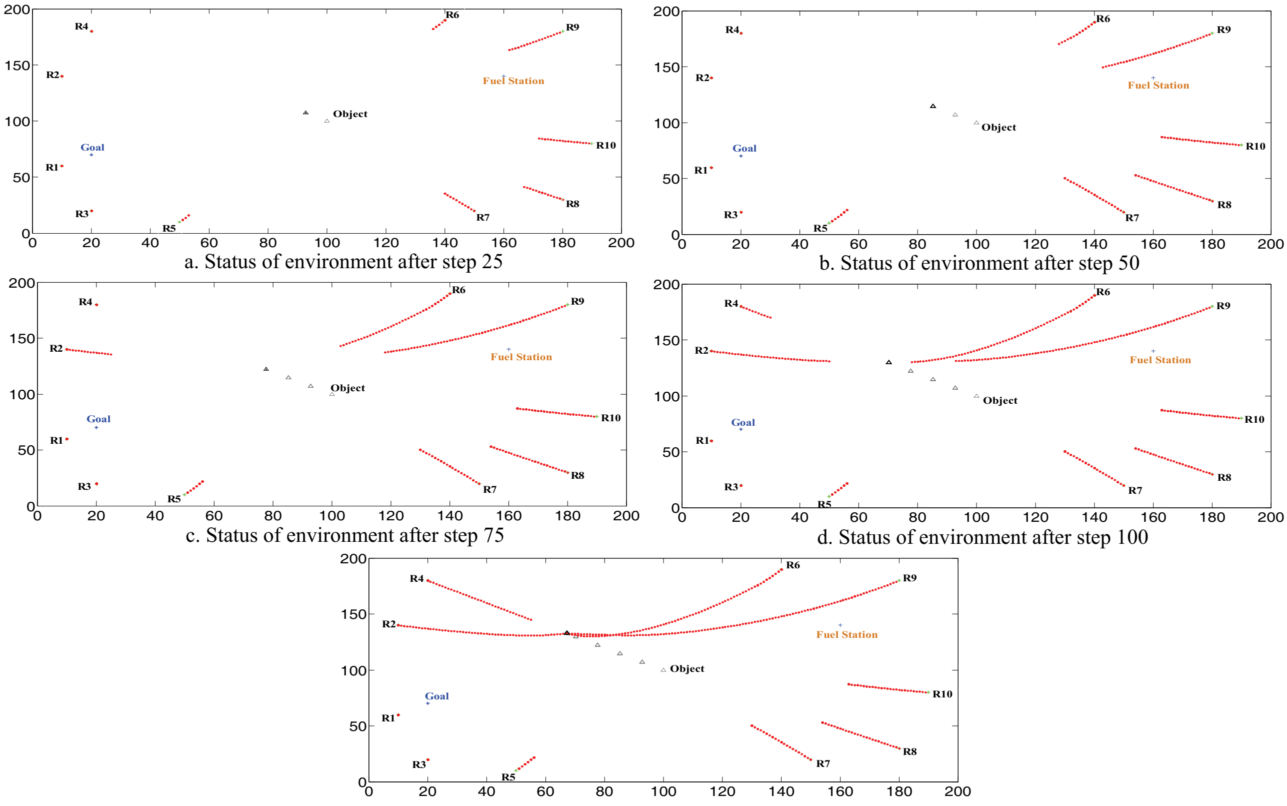
b. Status of environment after step 50

c. Status of environment after step 75

d. Status of environment after step 100

e. Status of environment after step 125

Fig. 8. Five snapshots of single-object foraging with random motion of the object.



a. Status of environment after step 25

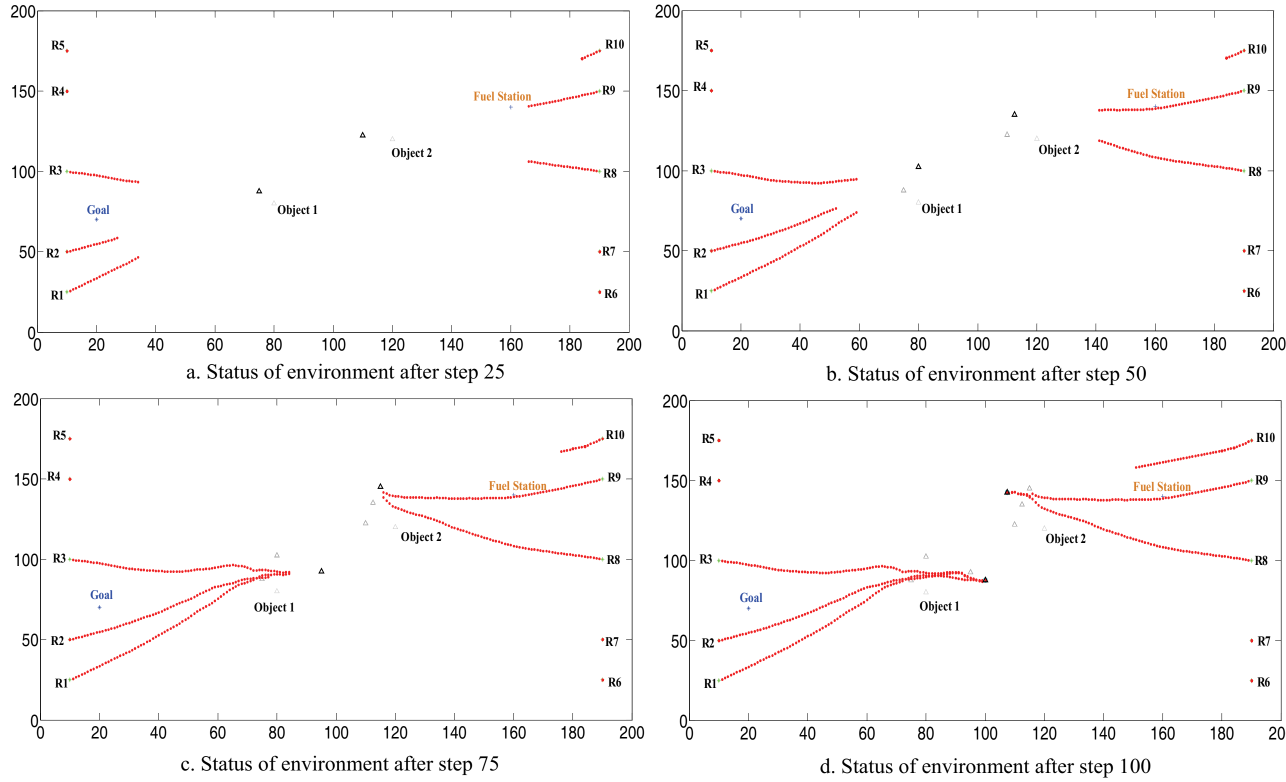
b. Status of environment after step 50

c. Status of environment after step 75

d. Status of environment after step 100

e. Status of environment after step 125

Fig. 9. Five snapshots of single object foraging with straight motion of the object.



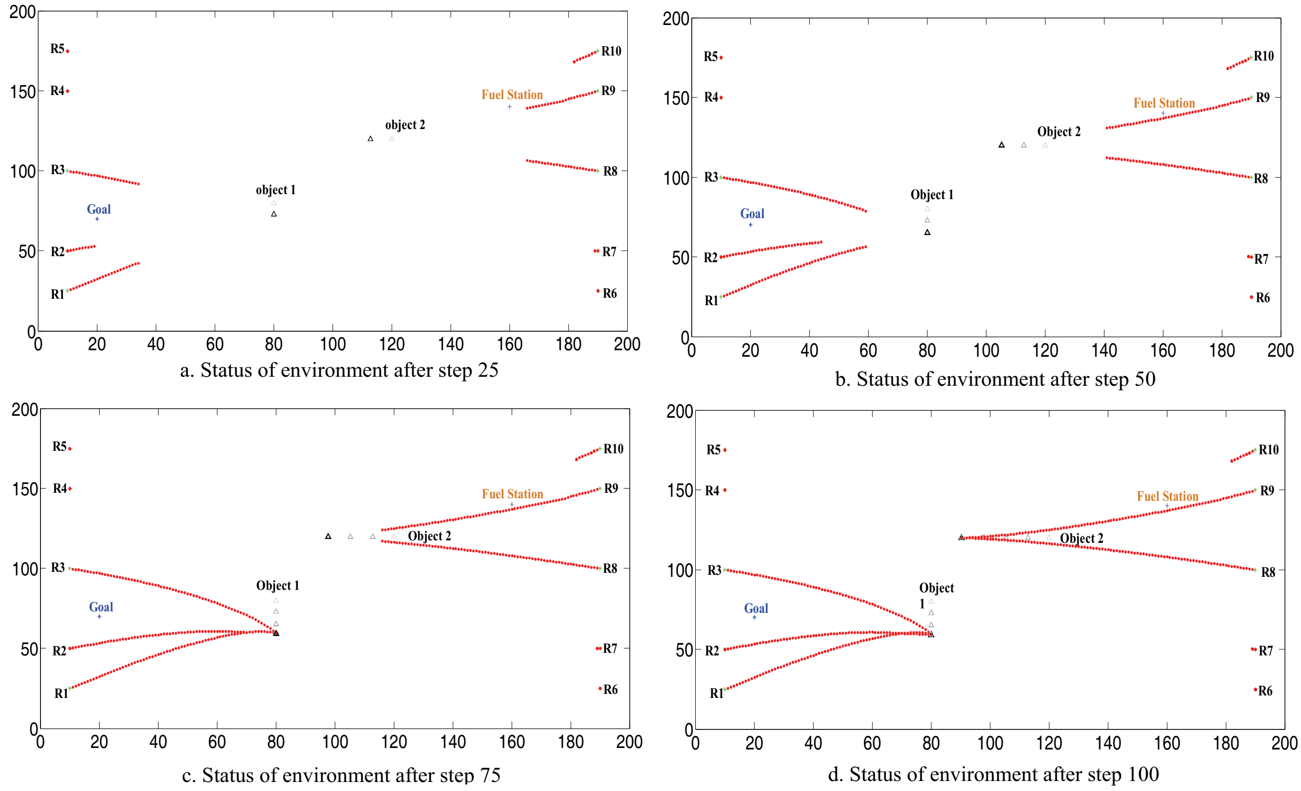
a. Status of environment after step 25

b. Status of environment after step 50

c. Status of environment after step 75

d. Status of environment after step 100

Fig. 10. Four snapshots of two-object foraging with random motion of the objects.



a. Status of environment after step 25

b. Status of environment after step 50

c. Status of environment after step 75

d. Status of environment after step 100

Fig. 11. Four snapshots of two-object foraging with straight motion of the objects.

$$N(R)=\sum_{r=0}^{R} r\mathrm{P}(r) \tag{5}$$

Now let S_a (which is proportional to A_a) be average steps that the robots inside achievable area should take to reach to the object; and S_g and S_f be the steps required for a robot to reach from the object to the goal and from the goal to the fuel station respectively. Obviously, the robots outside the achievable area will take no step. Then, the expected average steps taken by all robots to reach the object is given as:

$$S(R)=(S_a+S_g+S_f)\frac{N(R)}{R} \tag{6}$$

Finally we can obtain expected average fuel consumption by robots as:

$$F(R)=\left(E_f(S_a+S_f)+S_g\sum_{r=1}^{R}\frac{E_c\mathrm{P}(r)}{r}\right)\frac{N(R)}{R} \tag{7}$$

Recall that E_f and E_c are fuel consumption rate of robots while carrying or not carrying an object.

Above discussion can be easily employed in multi object environments by extending equation (3) using inclusion-exclusion principle. For example in a two object environment (see Figure 12.b) we have:

$$\mathrm{P}=\mathrm{P}(A_{a1})+\mathrm{P}(A_{a2})-\mathrm{P}(A_{a1} \textbf{ and } A_{a2}) \tag{8}$$

Then aforementioned expected average numbers is obtained using equations (4) to (7).

6.2.2. *Comparative simulations*

In this sub-section we are going adopt a Monte Carlo approach to evaluate performance of the proposed method in comparison with following local heuristics introduced in section 2.1:

CT: Each robot selects closest achievable object to itself.

MSMCT: For achievable objects, each robot calculates distances from the robot itself to the object, from the object to the goal and from the goal to the fuel station. Then it selects the object that corresponds to the minimum sum.

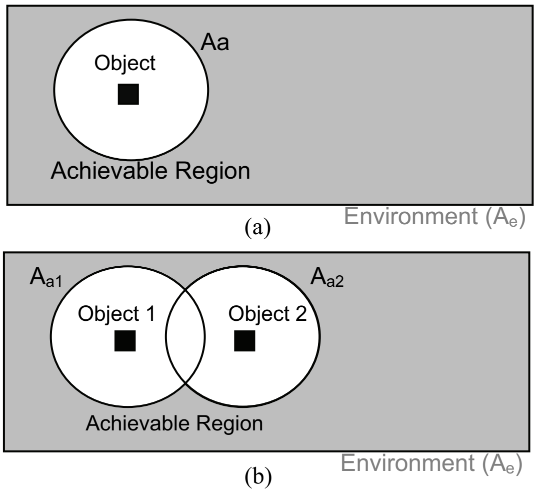


Fig. 12. Achievable regions in:
(a) single-object (b) two-object environments

We do not include MCT and MPT heuristics in this study as MSMCT is extension of MCT and in MPT the robots needs to know the positions of other robots which is not the case in our study.

To evaluate the performance of the algorithm using Monte Carlo approach, we considered an environment with a 200×200*m* area. Positions of the objects, goal and fuel station are shown in Table 9. The rate of fuel consumptions with and without carrying the object is assumed E_f=0.7 (unit per meter) and E_c=10 (unit per meter) respectively. The initial fuel of robots is assumed to be 350 units *(*F_i=350 i=1,...,R*)*. Available fuel at station is set to 900.

In first set of simulations a single-object problem was considered where only object 1 (see Table 9) exists in the environment. We run proposed artificial capital market (ACM), CT and MSMCT methods for R=5, ..., 10 robots in the environment. For each method simulations had run 50 times with random selection of initial positions of the robots. The results of average number of robots contributing in foraging, average step taken by robots, average fuel consumption of robots and average number of transported objects are shown in Figure 13. The expected average values obtained by equations (5), (6), (7) are also depicted. By increasing the number of robots probability of being in achievable region is increased, hence the number of robots

Table 9. Positions of goal, objects and fuel station in the environment considered in numerical analysis.

Goal position $\mathbf{P}_G=(X_G,Y_G)$	Position of Object 1 ($\mathbf{P}_{O1}$)	Position of Object 2 ($\mathbf{P}_{O2}$)	Fuel station position $\mathbf{P}_F=(X_F,Y_F)$
(20,70)	(100, 60)	(80, 160)	(160,140)

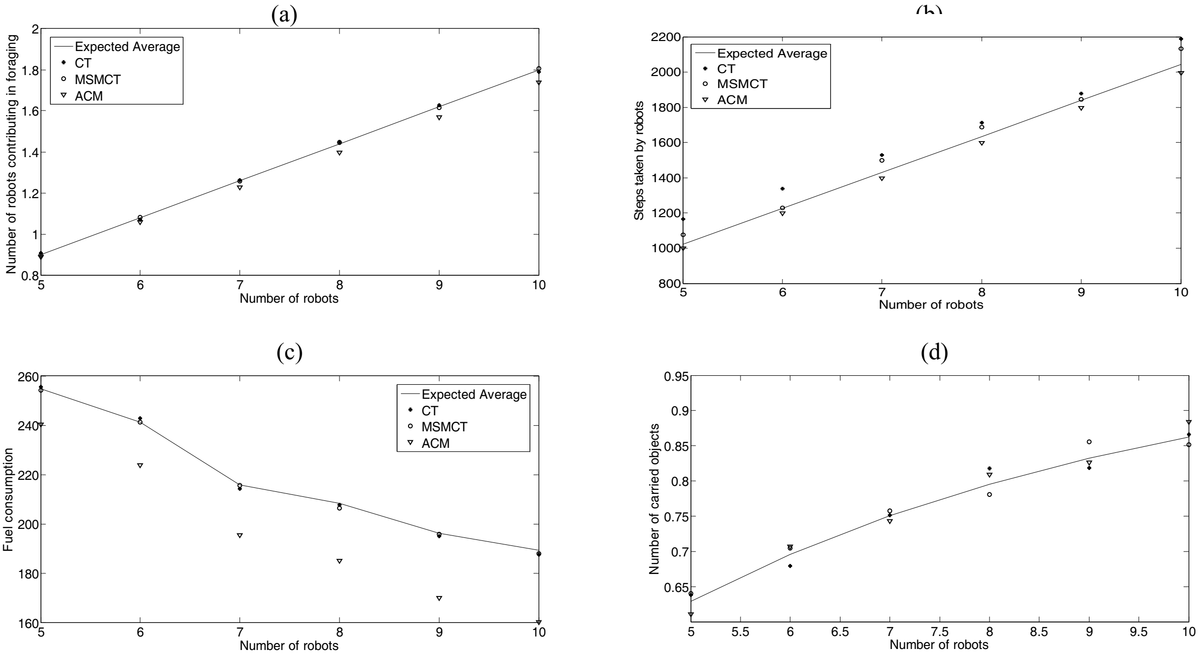


Fig. 13. Monte Carlo simulations of a single-object system. Simulations had run for *R*=5, ...,10 robots; each for 50 times with random selection of initial position of robots.

The average results of the proposed algorithm, ACM, is compared with the average results of CT and MSMCT heuristics in terms of (a) Number of robots contributing in foraging, (b) Steps taken by robots, (c) Fuel consumption, (d) Number of carried objects

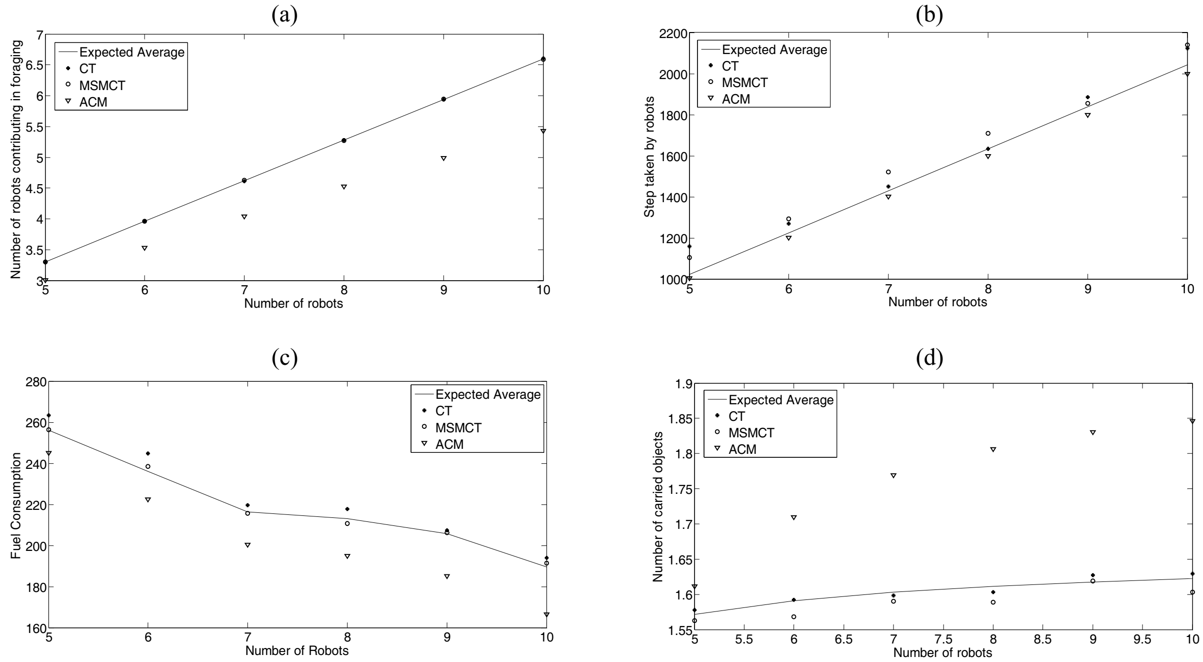


Fig. 14. Monte Carlo simulations of a two-object system. Simulations had run for *R*=5, ...,10 robots; each for 50 times with random selection of initial position of robots.

The average results of the proposed algorithm, ACM, is compared with the average results of CT and MSMCT heuristics in terms of (a) Number of robots contributing in foraging, (b) Steps taken by robots, (c) Fuel consumption, (d) Number of carried objects contributing in foraging and the number of steps taken by robots are increased (Figures 13.a and 13.b), average fuel consumption is decreased (Figure 13.c) and the probability of the object to be carried tends to one (Figure 13.d). In single-robot problem, CT and MSMCT are equivalent and their results are around expected average curve in all sub-figures of Figure 13. In CT, MSMCT and expected average approaches, if the robot is located in achievable region, it will definitely contribute in carrying the object. Nevertheless, in the proposed ACM method some robots inside achievable region may decide not to contribute in carrying the object. Therefore, in ACM less robots contribute in foraging, resulting less steps and less fuel consumption (See Figure 13. a, b, c). There is no meaningful difference between results of ACM and other methods in Figure 13.d since only one object exists in the environment.

In the second set of simulations, the same approach was adopted for two-object problems and the results of running ACM, CT, MSMCT methods (each 50 times) for R=5, …, 10 robots are illustrated in Figure 14. The overall analysis provided for single-object environment is true here except that some performance indices of CT and MSMCT are not the same. About number of robots those contribute in foraging there is no difference between CT and MSMCT since the robots inside achievable region will definitely contribute in carrying at least one object. In CT, a robot selects the nearest object and it does not take in account distances from the objects to goal and fuel station. Thus the robots are likely to cover more distances which results in more steps and more fuel consumption. In MSMCT, the robots consider distances from the objects to goal and then to fuel station, hence their taken steps and fuel consumption are less than CT. One drawback of MSMCT is that all robots inside the achievable region have to contribute in foraging while in the proposed ACM method the robots may decide to not contribute in the task. On the other hand in ACM, a robot considers decisions of other robots to make a better decision. Consequently number of steps and fuel consumption in ACM is less than both CT and MSMCT. Finally average number of carried objects in ACM is more. The reason is that in CT/MSMCT the robots choose the closest/most starved tasks and if one of the objects are closest/most starved to all robots, then the other object will selected by no robot. Therefore in ACM, two objects have more chance to be carried by robots.

7. Conclusions

Multi-robot multi-object foraging problem with dynamic objects is a complicated problem that can be a good test bed for multi agent problems. Artificial capital market as a new variant of market mechanisms was introduced and employed for task allocation in a multi-robot foraging system. In artificial capital market the robots are considered as investors those try to increase their outcomes by investing on some bundles of assets. The assets are sub-tasks of transporting of objects. Components of the market were defined and negotiation algorithm was proposed. The negotiation algorithm consists of a subroutine for decision making in static markets which is employed as a core routine for decision making in dynamic markets.

To evaluate performance of the algorithm two set of analysis were conducted including qualitative and numerical analysis, each supported by simulations in MATLAB. Qualitative analysis showed that the algorithm is robust and fast and the solutions are near optimal in the environment with static objects. As decision making in static market is the core of decision making in dynamic market, it can be concluded that the solutions obtained for the environments with dynamic objects would be robust and near optimal as well. In numerical analysis a Mont Carlo approach was adopted to study the performance of the algorithm in environments with different number of robots. The results were compared with the results of earlier task allocation heuristics namely closest task first (CT) and most starved-most closest task first (MSMCT) heuristics. The analysis showed that the proposed algorithm is superior in terms of four criteria including average number of robots contributing in foraging, average steps taken by robots, average fuel consumption and average carried objects.

References

1. A. Winfield, Foraging robots, In Meyers, R., ed. Encyclopaedia of Complexity and Systems Science. Springer, New York, pp. 3682-3700,
2. Y. Litus, R. T. Vaughan, , P. Zebrowski, "The frugal feeding problem: Energy-efficient, multi-robot, multi-place rendezvous", In Proc. of IEEE International Conference on Robotics and Automation, 2007, pp. 27-32., April 2007, doi: 10.1109/ROBOT.2007.363760
3. W. Liu, A. Winfield, J. Sa, J. Chen, L. Dou, "Strategies for energy optimization in a swarm of foraging robots". In Swarm Robotics, pp. 14-26, Springer Berlin Heidelberg, 2007, doi: 10.1007/978-3-540-71541-2_2
4. T. Stirling, D. Floreano, *Energy efficient swarm deployment for search in unknown environments*, In Swarm Intelligence, Springer Berlin Heidelberg, pp. 562-563, 2010, doi: 10.1007/978-3-642-15461-4_61

5. J. H. Lee, H. T. Kim, C. W. Ahn, "Foraging swarm robots system adopting honey bee swarm for improving energy efficiency" In Proc. of the 6th International Conference on Ubiquitous Information Management and Communication, p 100, February 2012**,** doi: 10.1145/2184751.2184867
6. S. Kernbach, V. A. Nepomnyashchikh, T. Kancheva, O. Kernbach, "Specialization and generalization of robot behaviour in swarm energy foraging", Mathematical and Computer Modelling of Dynamical Systems, 18(1), 131-152, 2012, doi: 10.1080/13873954.2011.601421
7. M. Bennewitz , W. Burgard, S. Thrun, "Optimizing schedules for prioritized path planning of multi-robot systems". In Proc. of IEEE International Conference on Robotics and Automation, vol. 1, pp. 271-276., 2001, doi: 10.1109/ROBOT.2001.932565
8. Y. Guo, L. E. Parker, "A distributed and optimal motion planning approach for multiple mobile robots". In Proc. of IEEE International Conference on Robotics and Automation (ICRA), vol. 3, pp. 2612-2619, 2002,
9. S. Liu, L. Mao, J. Lu., "Path planning based on ant colony algorithm and distributed local navigation for multi-robot systems", In Proc. of IEEE International Conference on Mechatronics and Automation, pp. 1733-1738, June 2006, doi: 10.1109/ICMA.2006.257476
10. L. E. Parker, *Path planning and motion coordination in multiple mobile robot teams*, Encyclopedia of Complexity and System Science, Springer, Heidelberg, 2009
11. A. Akbarimajd, A. Hassanzadeh, "Autonomously Implemented Versatile Path Planning for Mobile Robots Based on Cellular Automata and Ant Colony", International Journal of Computational Intelligence Systems, 5(1), 39-52, 2012
12. K. Solovey, D. Halperin, *k-Color Multi-Robot Motion Planning*. Algorithmic Foundations of Robotics X, pp. 191-207,Springer Berlin Heidelberg, 2013, doi: 10.1007/978-3-642-36279-8_12
13. N. Hoff, R. Wood, R. Nagpal, “Distributed Colony-Level Algorithm Switching for Robot Swarm Foraging”, Distributed Autonomous Robotic Systems Springer Tracts in Advanced Robotics Volume 83, 2013, pp 417-430, doi: 10.1007/978-3-642-32723-0_30
14. G. A. Kaminka, D. Erusalimchik, S. Kraus, "Adaptive multi-robot coordination: A game-theoretic perspective" In Proc. 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 328-334, May 2010, doi: 10.1109/ROBOT.2010.5509316
15. N. Lau, L. S. Lopes, G. Corrente, N. Filipe, R. Sequeira, "Robot team coordination using dynamic role and positioning assignment and role based setplays", Mechatronics, 21(2), 445-454, 2011, doi: 10.1016/j.mechatronics.2010.05.010
16. M. Lewis, H. Wang, S. Y. Chien, P. Scerri, P. Velagapudi, K. Sycara, B. Kane, "Teams organization and performance in multi-human/multi-robot teams". In Proc. IEEE International Conference on Systems Man and Cybernetics (SMC), pp. 1617-1623, October 2010, doi: 10.1109/ICSMC.2010.5642379
17. K. Sugawara, T. Kazama, T. Watanabe, "Foraging behavior of interacting robots with virtual pheromone" In Proc. of International Conference on Intelligent Robots and Systems (IROS), Vol. 3, pp. 3074-3079, October 2004, doi: 10.1109/IROS.2004.1389878
18. E. Klavins, "Communication complexity of multi-robot systems", Algorithmic Foundations of Robotics V, pp. 275-292, Springer Berlin Heidelberg, 2004, doi: 10.1007/978-3-540-45058-0_17
19. J. Pugh, A. Martinoli, "Relative localization and communication module for small-scale multi-robot systems" In Proc. IEEE International Conference on Robotics and Automation ICRA, pp. 188-193, May 2006, doi: 10.1109/ROBOT.2006.1641182
20. P. Rybski, A. Larson, H. Veeraraghavan, M. LaPoint, M. Gini, "Communication strategies in multi-robot search and retrieval: Experiences with mindart", In Distributed Autonomous Robotic Systems 6, pp. 317-326, Springer Japan, 2007, doi: 10.1007/978-4-431-35873-2_31
21. C. Ampatzis, E. Tuci, V. Trianni, M. Dorigo, "Evolution of signaling in a multi-robot system: Categorization and communication" Adaptive Behavior, 16(1), pp. 5-26, 2008
22. N. R. Hoff, A. Sagoff, R. J. Wood, R. Nagpal, "Two foraging algorithms for robot swarms using only local communication", Proc. of IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 123-130, December 2010, doi: 10.1109/ROBIO.2010.5723314
23. J. H. Jung, S. Park, S. L. Kim, "Multi-robot path finding with wireless multihop communications", Communications Magazine, IEEE, 48(7), pp. 126-132, 2010
24. C. Agüero, M. Veloso, "Transparent multi-robot communication exchange for executing robot behaviors", In Highlights on Practical Applications of Agents and Multi-Agent Systems, pp. 215-222, Springer Berlin Heidelberg, 2012, doi: 10.1007/978-3-642-28762-6_26
25. G. Pini, M. Gagliolo, A. Brutschy, M. Dorigo, and M. Birattari, “Task partitioning in a robot swarm: a study on the effect of communication”. Swarm Intelligence, 7(2-3), pp. 1-27, 2013, doi: 10.1007/s11721-013-0078-7
26. K. N. Krishnanand, D. Ghose, "Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications", Multiagent and Grid Systems, 2(3), pp. 209-222, 2006
27. J. Pugh, A. Martinoli, "Inspiring and modeling multi-robot search with particle swarm optimization", In IEEE Swarm Intelligence Symposium (SIS), pp. 332-339, April 2007, doi: 10.1109/SIS.2007.367956
28. A. Campo, M. Dorigo, "Efficient multi-foraging in swarm robotics", In Advances in Artificial Life, pp. 696-705, Springer Berlin Heidelberg, 2007, doi: 10.1007/978-3-540-74913-4_70
29. M. A. Garcia, O. Montiel, O. Castillo, R. Sepúlveda, P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation", Applied Soft Computing, 9(3), pp. 1102-1110, 2009, doi: 10.1016/j.asoc.2009.02.014
30. W. Liu, A. F. Winfield, "Modeling and optimization of adaptive foraging in swarm robotic systems", The

International Journal of Robotics Research, 29(14), pp, 1743-1760, 2010, doi: 10.1177/0278364910375139

31. F, Arvin, K. Samsudin, A. R. Ramlib and M. Bekravi, “Imitation of Honeybee Aggregation with Collective Behavior of Swarm Robots”, International Journal of Computational Intelligence Systems, 4(4), pp. 739-748, 2011, doi: 10.1080/18756891.2011.9727825
32. T. S. Dahl, M. J. Mataric, G. S. Sukhatme, "Multi-robot task-allocation through vacancy chains" In Proc. IEEE International Conference on Robotics and Automation, Vol. 2, pp. 2293-2298. September 2003, doi: 10.1109/ROBOT.2003.1241935
33. M. J. Krieger, J. B. Billeter, L. Keller, "Ant-like task allocation and recruitment in cooperative robots". Nature, 406(6799), 992-995, 2000
34. P. Dasgupta, "Multi-robot task allocation for performing cooperative foraging tasks in an initially unknown environment", In Innovations in Defence Support Systems-2, pp. 5-20, Springer Berlin Heidelberg, 2011, doi: 10.1007/978-3-642-17764-4_216
35. B. P. Gerkey, M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems", The International Journal of Robotics Research, 23(9), 939-954, 2004, doi: 10.1177/0278364904045564
36. K. Lerman, C. Jones, A. Galstyan, M. J. Matarić, "Analysis of dynamic task allocation in multi-robot systems", The International Journal of Robotics Research, 25(3), 225-241, 2006, doi: 10.1177/0278364906063426
37. T. H. Labella, M. Dorigo, J. L. Deneubourg, "Self-organised task allocation in a group of robots" In Distributed Autonomous Robotic Systems 6, pp. 389-398, Springer Japan, 2007, doi: 10.1007/978-4-431-35873-2_38
38. N. Michael, M. M. Zavlanos, V. Kumar and G. J. Pappas, “Distributed Multi-Robot Task Assignment and Formation Control,” in Proc. of the IEEE Conference on Robotics and Automation, Pasadena, CA, pp.128–133, May 2008., doi:10.1109/ROBOT.2008.4543197
39. M. J. Matarić, G.S. Sukhatme, E. H. Østergaard, "Multi-robot task allocation in uncertain environments". Autonomous Robots, 14(2-3), 255-263, 2003
40. R. D. Morton, G. A. Bekey, C. M. Clark, "Altruistic task allocation despite unbalanced relationships within multi-robot communities", In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, pp. 5849-5854, October, 2009, doi: 10.1109/IROS.2009.5354072
41. C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, B. Brambilla and M. Dorigo, "Self-organized task allocation to sequentially interdependent tasks in swarm robotics", Autonomous Agents and Multi-Agent Systems, 5, 305-327, 2012
42. M. Kloetzer, A. Burlacu, D. Panescu, "On a Class of Multi-Robot Task Allocation Problems", In Information Control Problems in Manufacturing, Vol. 14, No. 1, pp. 841-846, May 2012
43. M. X. Yuan, Y. F. Jiang, Y. Shen, Z. L. Ye, Q. Wang, "Task Allocation of Multi-Robot Systems Based on a Novel Explosive Immune Evolutionary Algorithm", Applied Mechanics and Materials, 246, 331-335, 2013, doi: 10.4028/www.scientific.net/AMM.246-247.331
44. X. Zheng and S. Koenig, "Negotiation with Reaction Functions for Solving Complex Task Allocation Problems", The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems St. Louis, USA, October 11-15, 2009, doi:10.1109/IROS.2009.5354326
45. H. S. Dinga, P. Heb, F. X. Wua and Y. H. Sun, “The impact of cooperation and competition on the performance of flexible decentralized manufacturing networks”, International Journal of Computational Intelligence Systems, 6(1), pp. 174-185, 2013, doi: 10.1080/18756891.2013.761777
46. M. Ljesnjanin, J. Velagic, "A Market Based Approach for Complex Task Allocation for Wireless Network Based Multi-Robot System", XXII International Symposium on Information, Communication and Automation Technologies, 2009. ICAT 2009, Bosnia, Oct. 2009, doi:10.1109/ICAT.2009.53484283.
47. M. Dias and A. Stentz. "Opportunistic optimization for market-based multirobot control". In Proc. of the International Conference on Intelligent Robots and Systems, pp 2714–2720, 2002., doi:10.1109/IRDS.2002.1041680
48. 4. M. Dias, R. Zlot, N. Kalra, and A. Stentz. "Market-based multirobot coordination: A survey and analysis". Proceedings of the IEEE, 94(7):1257–1270, 2006., doi:10.1109/JPROC.2006.876939
49. S. Koenig, C. Tovey, X. Zheng, and I. Sungur. "Sequential bundle-bid single-sale auction algorithms for decentralized control". In Proceedings of the International Joint Conference on Artificial Intelligence, pages 1359–1365, 2007.
50. M. B. Dias. TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments. PhD thesis, Robotics Institute, Carnegie Mellon University, 2004.
51. R. Nissim, and R. Brafman, “Distributed Heuristic Forward Search for Multi-Agent Systems”, arXiv preprint arXiv:1306.5858, 2013
52. M. Al-Asfoor, B. Neville and M. Fasli, “Heuristic resource search in a self-organised distributed multi agent system.” In Self-Organizing Systems, vol. 7166, pp. 84-89. Springer Berlin Heidelberg, 2012, doi: 10.1007/978-3-642-28583-7_8
53. D. J. Farber and K. C. Larson. "The structure of a distributed computing system". In Proceedings of the Symposium on Computer-Communications Networks and Teletraffic, 1972.
54. E. G. Jones, B. Browning, M. B. Dias, “Dynamically Formed Heterogeneous Robot Teams Performing Tightly-Coordinated Tasks,” in Proc. of the IEEE Conference on Robotics and Automation, Orlando, Florida, pp. 570-575, May 2006., doi:10.1109/ROBOT.2006.1641771
55. E. G. Jones, B. Browning, M. B. Dias, “Dynamically Formed Heterogeneous Robot Teams Performing Tightly-Coordinated Tasks,” in Proc. of the IEEE Conference on Robotics and Automation, Orlando, Florida, pp. 570-575, May 2006., doi:10.1109/ROBOT.2006.1641771
56. M. Nanjanath, M Gini, “Auctions for task allocation to robots", In Proc. of the 9th International Conference on Intelligent Autonomous Systems,” Tokyo, pp. 550–557, 2006., doi:10.1109/ROBOT.2006.1642122

57. P. Garcia, P. Caama˜no, F. Bellas, and R. J. Duro, “A Behavior Based Architecture with Auction-Based Task Assignment for Multi-robot Industrial Applications,” Lecture Notes in Computer Science: Bioinspired Applications in Artificial and Natural Computation, Part II, LNCS 5602, pp. 372–381, 2009., doi:10.1007/978-3-642-02267-8_40
58. K. Zhang, E. G. Collins, D. Shi, X. Liu, and O. Chuy, “A stochastic clustering auction for centralized and distributed task allocation in multi-agent teams”, Distributed Autonomous Robotic Systems, Vol 8, pages 345–354, Tsukuba, Ibaraki, Japan, 2008., doi:10.1007/978-3-642-00644-9_31
59. K. Zhang, E. G. Collins and A. Barbu, “Efficient Stochastic Clustering Auctions for Agent-Based Collaborative Systems”, IEEE International Conference on Robotics and Automation (ICRA 2011), Agent Technology in Robotics and Automation Workshop , Shanghai, China, May 13, 2011
60. C. Raoab and X. Wub, “Optimization Decision Method for Procuring Divisible Goods”, International Journal of Computational Intelligence Systems, 5(6), pp. 1160-1172, 2012, doi: 10.1080/18756891.2012.747682
61. M. M. Ortiz, J. de Lope Asiaín and F. de la Paz, “Auction based method for graphic-like maps inspection by multi-robot system in simulated and real environments”, Robotics and Autonomous Systems, 61(7), pp. 676-681, 2013, doi: 10.1016/j.robot.2012.08.007
62. R. G. Smith, “The contract net protocol: High-level communication and control in a distributed problem solver,” IEEE Trans. Comput., vol. C-29, pp. 1104, Dec. 1980., doi:10.1109/TC.1980.1675516
63. H. V. D. Parunak, “Manufacturing experience with the contract net”, Distributed artificial intelligence, 1, 285-310, 1987
64. T. Sandholm, “Contract types for satisficing task allocation”, In Proceedings of the AAAI spring symposium: Satisficing models, pp. 23-25, 1998
65. Z. Juhasz, P. Paul, “Scalability analysis of the contract net protocol” In 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 346-346, May 2002, doi: 10.1109/CCGRID.2002.1017157
66. T. Lemaire, R. Alami, S. Lacroix, “A distributed tasks allocation scheme in multi-UAV context. In Robotics and Automation” In Proc. of IEEE International Conference on Robotics and Automation (ICRA), vol. 4, pp. 3622-3627, April 2004, doi:
67. F. S. Hsieh, “Analysis of contract net in multi-agent systems”. Automatica, vol. 42 No. 5, pp. 733-740, 2006, doi: 10.1016/j.automatica.2005.12.002
68. D. Sun, J. Wu, "Multi-agent Coordination Based On Contract Net Protocol", 2009 International Symposium on Intelligent Ubiquitous Computing and Education, Washington, DC, USA, 2009, doi:10.1109/IUCE.2009.111
69. P. Sapra, M. Memoria, “Multi-Agent Systems for Adaptive and Efficient Job Scheduling Service in Grids”. International Journal of Computer Applications, vol. 1, No 9, pp. 26-30, 2010
70. H. Lili and Y. Huizhen, “Improvement and Simulation of Contract-Net-Based Task Allocation for Multi-robot System”. In Proceedings of the 2011 2nd International Congress on Computer Applications and Computational Science, pp. 61-67, Springer Berlin Heidelberg, January 2012, doi: 0.1007/978-3-642-28314-7_8
71. X. Li, Z. Dong, H. Wang, Q. Meng, Z. Wang, and Y. Zhang, “A novel approach to selecting contractor in agent-based multi-sensor battlefield reconnaissance simulation”. International Journal of Computational Intelligence Systems, 5(6), pp. 985-995, 2012, doi: 10.1080/18756891.2012.747652
72. M. P. Wellman, "A market-oriented programming environment and its application to distributed multicommodity flow problems", arXiv preprint cs/9308102, 1993
73. M. P. Wellman, and J. Hu, "Conjectural equilibrium in multiagent learning", Machine Learning, 33(2-3), 179-200, 1998,
74. T. Kaihara, "Multi-agent based supply chain modelling with dynamic environment", International Journal of Production Economics, 85(2), 263-269, 2003, doi: 10.1016/S0925-5273(03)00114-2
75. O. Kwon, "Multi-agent system approach to context-aware coordinated web services under general market mechanism". Decision Support Systems, 41(2), 380-399, 2006, doi: 10.1016/j.dss.2004.07.005
76. G. Simzan, A. Akbarimajd, M. Khosravani, "A market based distributed cooperation mechanism in a multi-robot transportation problem", In Proceeding of Int. Conf. Intelligent Systems Design and Applications (ISDA), 22-Cordoba, Spain, November 2011, doi:10.1109/ISDA.2011.6121621