Cooperative Problem Solving Using an Agent-Based Market

David Cornforth and Michael Kirley

School of Environmental and Information Sciences
Charles Sturt University, Albury, NSW, Australia
dcornforth@csu.edu.au

Department of Computer Science and Software Engineering.
University of Melbourne, Melbourne, Vic, Australia
mkirley@cs.mu.oz.au

Abstract. A key problem in multi-agent systems research is identifying appropriate techniques to facilitate effective cooperation between agents. In this paper, we investigate the efficacy of a novel market-based aggregation technique in addressing this problem. An incremental transaction-based protocol is introduced where agents establish links by buying and selling from each other. Market transactions equate to agents coordinating their plans and sharing their resources to meet the global objective. An important contribution of this study is to clarify whether, in some circumstances, a market-based model leads to the effective formation of agent teams (or coalitions) and thus, solutions to the problem-solving task.

1 Introduction

An increasing number of computational systems may be viewed in terms of multiple, interacting autonomous agents. Interactions might include cooperation to achieve a joint goal, competition for resources, negotiation over a set of tasks to perform, or the buying or selling of resources [1]. If we adopt a game-theoretic perspective, the agents play a general-sum game in which particular coalitions or teams of agents working together have a higher utility (or relative fitness) than other agents in the population [2]. One way to view this coalition formation process is as a distributed search through the space of all possible configurations.

An alternative perspective favored by the Artificial Life community sees multi-agent systems as simulations based on metaphors inspired by ecological, economic or social communities. Here, the agents, their behavioral rules, and their mutual interactions define complex systems. Holland [3] suggests that the agents may be thought of as building blocks representing formal components of the model, built to understand the complex patterns of emergent behavior underlying the system. An inherent feature of these systems is the ability of the agent to group together to form composite entities, also known as modules, clusters, teams or coalitions, depending on the terminology of the respective discipline. However, it is still an open question as to

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 60–71, 2004.

how agents in a complex system form coalitions, and how these coalitions self-organize into hierarchies.

In this paper, we begin to address this question by focusing on cooperation in multi-agent systems. Specifically, we are interested in a situation such that given a specific goal, which cannot be satisfied by a single agent, a collective effort by several agents is required. In this instance, agents must coordinate their plans and share their resources to meet the global objective. We propose a novel economic market-based mechanism to facilitate cooperation in multi-agent systems. Using a highly idealized model, we illustrate how agents in a system can use a series of incremental transactions to form appropriate teams or coalitions for "solving" a given problem. Here, the multi-agent system may be viewed as a virtual market place populated by heterogeneous self-interested traders (agents) attempting to maximize their own utility. Buyers try to trade at the lowest possible price. Conversely, sellers try to trade at the highest price possible. Successful transactions represent steps in a bottom-up decentralized team formation protocol.

This study parallels (a) research into coalition formation protocols – where rational agents negotiate to join agent teams; (b) computational synthesis research – where low-level building blocks or features are combined to achieve given arbitrary high-level functionality in multi-agent systems, and (c) artificial symbiotic processes research – where alternative aggregation mechanisms based on a mutualism metaphor are used. An important contribution of this work is to clarify whether, in some circumstances, a market-based model leads to the effective formation of agent coalitions and thus, solutions to the problem solving task. To meet this objective, a number of simulations are presented focusing on the effectiveness of the aggregation process. In addition, we explore suitable mechanisms for fostering and maintaining diversity within the agent population.

The remaining sections of the paper are organized as follows. Related work and background material is presented in Section 2. In Section 3, the market-based model is described. Section 4 illustrates the functionality of the model using a pattern recognition task. In Section 5, we present the simulation results. We conclude with a discussion of the results and the implications of this work.

2 Background and Related Work

2.1 Coalitions and Cooperation

Cooperation is a key process in many multi-agent systems. Agents may cooperate to collectively solve some problem or perform some task, where a single agent could not succeed. In this scenario, each individual agent is able to carry out its tasks through interaction with a small number of *neighboring agents*. When interdependent problems arise, the agents in the systems must cooperate with one another to ensure that interdependencies are properly managed.

In computer science, cooperation has been studied extensively by Axelrod [4] and Huberman [5]. This work has been extended into the autonomous agent domain in the areas of auction theory [6], team formation [7] and coalition formation [8] [9] [10]

[11]. Much of this work has focused on how a group of agents make particular decisions and the associated utility values associated with the decisions. Sandholm and Lesser [12] present an interesting coalition formation process model for bounded-rational agents and a general classification of coalition games. They allow for varying coalition values, but provide the agents with heuristics that could be computed in polynomial time

The main question in every coalition formation application is how to determine which agents collaborate. While game theory is a useful analytical tool, Wooldridge and Jennings [13] suggest that it is not a good engineering tool primarily because of the type of representation employed by game theory. Wellman [14] suggest that agent interaction models employing market-based control mechanisms offer the possibility of fostering cooperation. It is this notion that provides some of the motivation for the model proposed in Section 3.

2.2 Artificial Symbiotic Models

Perhaps the most well known artificial symbiotic model is Potter and De Jong's [15] cooperative coevolutionary model. In this model, the artificial ecosystem consists of two (or more) species. Species (or modules) interact with one another within a shared domain model and have a cooperative relationship. Species evolve independently. However, the fitness value of an individual is directly related to how well that individual collaborated with representatives from each of the other species in solving the "super goal." Fundamentally, this model is a divide-and-conquer approach, where the system cycles between *decomposition – evolution – collaboration and evaluation*. In later work [16], the architecture was extended to include dynamic speciation and extinction. New species were added to the model based upon some measured stagnation in the evolutionary process. Other species were destroyed if they were no longer making significant contributions. This enhanced model was applied successfully to string covering problems and evolving cascade networks.

Watson and Pollack [17] have investigated how mechanisms based on abstract symbiotic processes affect adaptation in evolutionary systems. Specifically, they have developed algorithms where higher-level complexes are formed from simple "modules" or building blocks. This notion of building blocks is a fundamental principle of genetic algorithms. However, their "aggregation of modules" is directly related to coevolutionary interactions within the evolving population. The most important features of their model include: (a) techniques for combining modules based on symbiotic processes, (b) the introduction of explicit mechanisms ensuring that modules co-adapt to cover complementary parts of the problem domain, and (c) the use of appropriate techniques that can be used to determine the relative worth (fitness) of a module, including Pareto comparisons.

There are a number of similarities between each of the models described above. For instance, the flexibility inherent in Potter and De Jong's extended architecture and Watson and Pollack's hierarchical composition of modules have computational advantages. However, Daida and co-workers [18] have shown that many caveats exist either in adopting symbiosis as a computational heuristic, or in modeling symbiosis as an aspect of complex adaptive system behavior. They contended that in each case, symbiosis should be considered as a kind of operator instead of a state.

3 A Framework for Market-Based Problem Solving

The proposed market-based model is an incremental approach to problem-solving based on a bottom-up team (or coalition) formation protocol. The model consists of autonomous units – agents – interacting with each other as well as with an environment. At any time *t,* the system will contain a population of agents **A** = { A_1, A_2,…A_n}.

An agent refers to a localized entity with decision making capabilities. It can be a single individual or a coalition of individuals. In this instance, an agent represents a basic building block in the problem-solving task. Each agent has specific functionality and behaviors represented by the tuple $\mathbf{A}_i$ = < *resource,* P_{sell}, G_{fit}, P_{buy} L_{fit}> where: the *resource* represents the product encapsulated by the agent (which can be traded); P_{sell}, is the probability that the agent will offer its resource for sale; G_{fit} is the fitness gain – a weighting factor for the minimum selling price; P_{buy} is the probability that the agent will make a bid in the current market; and L_{fit} - the fitness loss – weighting factor for the maximum bid price.

We assume that all atomic agents are peers. That is, there is no default hierarchy among individual atomic agents. However, individual agents in the economy have heterogeneous beliefs concerning realization of possible outcomes.

The artificial market consists of a sequence of modified auctions. Randomly selected agents participate in a single auction and have the intention of buying the target resource from another agent. They maintain information about the resource they wish to purchase and their private valuation of this resource (the maximum amount that they are willing to pay for the desired item). A successful transaction, that is, the situation where agent A_i sells its resource to agent A_j establishes a trade-link or a coalition between the agents. Here, we use the term coalition to describe the team of agents drawn from A, who have worked together (traded-resources) to accomplish a task. In utility-theoretic terms, the utility (or fitness value) of each agent A_i is a function of G_{fit} and L_{fit} parameters of the individual agents.

To facilitate the functioning of the market, we have implemented a modified version of the Contract Net Protocol (CNET) [19]. CNET provides a general framework to describe negotiation processes between agents. Essentially, this protocol is based on a collection of agents, which cooperate in achieving sub-goals which, in turn collectively meet some high-level goal. CNET provides a means to find the "best" acquaintance for a given task. A key component of this protocol involves agents making decisions based on each agent's perspective of the current state of the world. The following steps encapsulate the basic functionality of the modified CNET protocol:

1. Task announcement and processing – corresponds to specifying the complete problem to be solved. On receipt of a task announcement, an agent decides if it is eligible for the task (or some part of the task). It does this by looking at the eligibility specifications contained in the announcement. If it is eligible (that is, the agent can solve some part of the problem), then details of the task are stored, and the agent will subsequently bid for the task.
2. Bid processing – agents who have responded to the task announcement bid to gain control of the selling agents resources. Details of the bid from would-be

contractors are stored by the would-be managers until the deadline for the task. The manager then awards the task to a single bidder.

3. Award processing – agents that bid for a task, but fail to be awarded it, simply delete details of the task. The successful bidder must attempt to expedite the task

It is important to note that the award processing phase may lead to different global utilities, depending upon who the successful agent/coalition is.

4 Problem Description

To illustrate the basic functionality of the trading model, a specific problem-solving task will be used – a string matching problem (pattern recognition of symbols). It is important to emphasize that we are interested in the general features of problem solving with agents that is applicable in a wide variety domains. However, we illustrate the phases of the market-based model using a concrete example.

A target string is drawn from a dictionary of English words longer than 16 letters. In this string-matching problem, the task of each agent is to synthesize the target string, but each agent is initialized with one letter (resource), and therefore must acquire other letters. All agents are able to buy or sell letters or word fragments to add to their collection. This task is not only a matter of permutation but also of the correct sequential composition, which in principle acts in parallel.

In Fig 1, we illustrate the outcome of repeated transactions within the market for the target string *acknowledgements*. The atomic agents – that is, agents encapsulating a single letter only – are shaded and can be found at the bottom of the hierarchy. When an agent purchases a letter or string fragment from another agent a coalition is formed. At the next level of the hierarchy the agent encapsulates the corresponding string.

As the model is iterated, agents trade with each other, buying and selling characters in exchange for a notional currency. After the model has been run, some agents will have

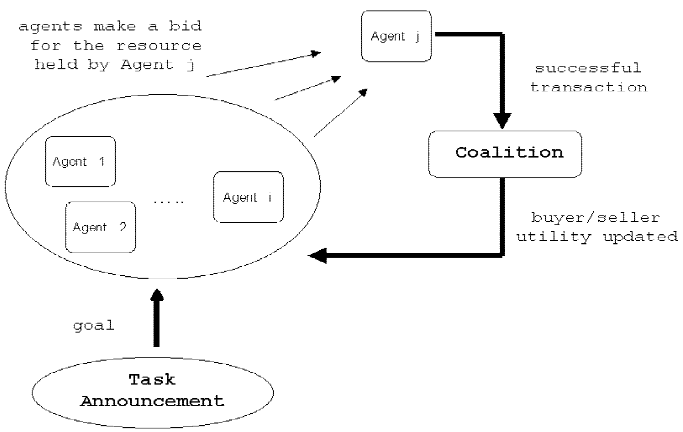


Fig. 1. An outline of the trading-model framework. The bidding-coalition formation phases are iterated until the goal is achieved (or a termination criterion is reached).

accumulated the correct letters to solve the problem , but they can still offer this complete collection for sale. Other agents will have no letters at all, having sold the letter they were initialized with. Other agents may have acquired partial solutions. All agents start with zero capital, but may gain or lose capital through trading. On each iteration of the model, one agent is selected at random to be the seller, and if its P_{sell} value is higher than a randomly generated number, the agent offers its letter or word fragment (coalition) for sale. If an agent has more than one letter, it cannot choose to offer some letters for sale and retain others: it offers all of its resources for sale. Other agents bid for these resources, as long as their P_{buy} value is higher than a randomly selected value. The price of any trade is determined by a simple formula that takes account of the number of letters that match the target word. Price setting is done using a tender system. The minimum selling price is determined by:

$$P_{\min} = m^2 G_{fit} \tag{1}$$

where P_{min} is the minimum selling price, m is the number of matches, and G_{fit} is the fitness gain. The bid price for buyer is determined by:

$$P_{bid} = (m_n^2 + (m_n m_c))L_{fit} \tag{2}$$

where P_{bid} is the bid price, m_n is the number of new matches gained from the purchase, m_c is the number of current matches the agent has, and L_{fit} is the fitness loss. P_{bid} takes into account of the fact that if the current number of matches is zero, the bid should be greater than zero, but if the number of matches gained by the purchase is zero, the bid price should also be zero. The buyer agent has the problem of calculating how to join the two strings together to maximize the resulting number of matches. This is solved by sliding the new string over the old and assessing the total number of matches at each position. The agent then chooses a position to maximize the number of matches. The new string always replaces letters of the existing string. This is illustrated in Table 1, where G_{fit} and $L_{fit} = 0.5$. Notice that the letter ‘l’ is duplicated, so that the original letter ‘l’ is overwritten by the new string in the best combination (22 matches).

The agent that produces the highest bid purchase the resource (characters), as long as $P_{bid} > P_{min}$. The sellers’ capital is increased by P_{bid} of the winning buyer, while the buyers’ capital is decreased by the same amount. The problem is solved when a copy of the target string is held by one of the agents.

It is possible for more than one solution to be found if there are enough letters (resources) in circulation among the agents. The solution is collaborative in the sense that agents must be willing to trade in order for the letters to pass into the control of a single agent. This means that the agent parameters (P_{sell}, G_{fit}, P_{buy}, and L_{fit}) must have values conducive to trading. The agents that have contributed their letters to this solution can be considered to be part of a coalition, since they have received payment, and thus contribute to the fitness value of the agent controlling the solution. Agents that trade are deemed to belong to the same coalition, while agents that do not trade with each other belong to different coalitions, or have no membership. After a number of iterations, these coalitions form a hierarchy supporting a solution, as illustrated in Fig. 2. The data in this figure were produced during an actual run of our model.

Table 1. Sample utility calculations. The new string is positioned against the target string to maximise the number of matches obtained.

String	**Resource**	**Calculations**
Target	`acknowledgements`	
Offered for sale	`ledge`	$P_{min} = 5^2 * 0.5 = 12.5$
Belonging to potential buyer	`acknowl`	

New combined string	`acknowlledge`
	`acknowledge`
	`acknoledge`
	
	`etc`
		$P_{bid} =$
		
		$(0^2 + (0*5))*0.5 = 0$
		$(4^2 + (4*7))*0.5 = 22$
		$(4^2 + (4*6))*0.5 = 20$

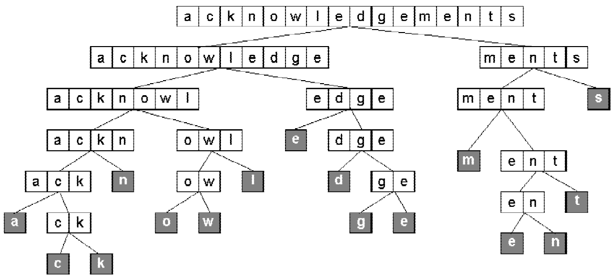


Fig. 2. Steps in the coalition formation process. Initially, all agents hold a single letter (resources). Agents then bid to gain control of additional letters to maximize their utility. In this diagram, we simply show the resource encapsulated by the agent at each level of the hierarchy for a given target word. The other agent parameters P_{Sell}, G_{fit}, P_{Buy}, and L_{fit} have been omitted for clarity.

5 Simulations

The market-based model has been designed as a continuous time, discrete-event simulation of a population of trading-agents. To examine the overall performance of model a number of simulation experiments were carried out. Specific parameters of interest include: the agent initialization states, behavioral rules (trading strategy), and the pattern of connectivity or possible interaction links.

5.1 Experiments

The aim of the first experiment was to establish the effectiveness and efficiency of the market-based model as a cooperative problem-solving tool. In particular, we are interested in determining how the agent/coalition capital levels (utility) are related to the number of correctly matching strings found.

The multi-agent system used in this simulation consisted of 1000 diverse agents. Here, diversity refers to the range of values that agent parameters are initialized with. The resource parameter of each agent was initialized with a randomly drawn letter from the English alphabet. Each of the other parameters – P_{sell}, G_{fit}, P_{buy}, and L_{fit} – was initialized with a random value from a uniform distribution between 0 and 1.0. These values do not change during a run of the model.

For a given target string, the market model was simulated for 5000 iterations. In this simulation, a total of 289 different target strings were considered – one run for each target string with 16 or more letters in the dictionary. As the agents were initialized with letters drawn from a uniform distribution, and as the distribution of letters in English words is far from uniform, the limit on the number of complete matches is the number of repeated letters in a single word. For example, it is common for the letter 'e' to be used 3 times in one word. This letter is expected to occur approximately 40 times (1000 / 26) in the initial population, so the maximum possible number of complete solutions is approximately 13 (40 / 3).

In the second experiment, we extend the simulation to include the possibility that some of the agents leave the market and new agents enter over the course of the run. This particular modification offers the possibility of: (a) promoting diversity across the agent populations and (b) mimicking real market places more closely. Determining which agents leave the market is not a straight forward problem. The utility of an agent (coalition) is a function of the number of matches obtained and its current capital. Agents are rewarded for matching characters in the solution, but punished for expending capital.

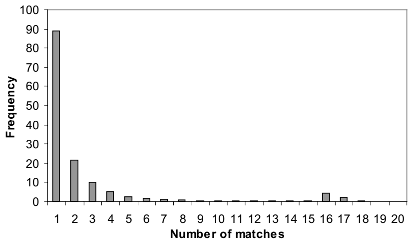
Consequently, we implement a form of "termination" based on an agents' utility. For example, if the termination threshold is set at 50%, then 50% of agents having zero matches are replaced, 50% of agents having one match are replaced, and so on until 50% of agents having a perfect match of the target string are replaced. When an agent is replaced, its string fragment is transferred to the new agent to minimize the loss of letters from the market. An exception is made for agents having a single letter that does not match the target string. These agents are replaced with a new agent having a letter drawn at random from a uniform distribution of letter in the alphabet. In either case, the new agents' capital is set to the average for agents with that number of matches. The remaining parameters are set in the same way as initialization.

Once again, this model configuration was executed for 5000 iterations for each of the target strings used in the first experiment.

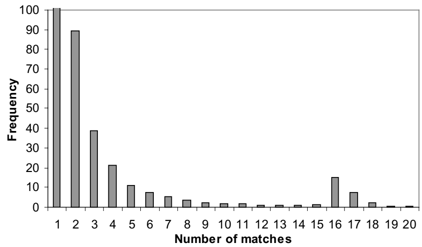
5.2 Results

Fig. 3 plots the frequency distribution of correct string matches averaged over all agents across the 289 runs. The results of experiments 1 and 2 are compared. It is interesting to note the improved performance in Fig 3b. This difference may be attributed to the fact that agents who failed to satisfy a predefined performance criterion were removed at regular time intervals, and replaced by randomly initialized agents. The result of the selection process is that some poorly performing agents are eventually replaced by fitter agents.

In Fig 4, we plot the number of matches *vs*. capital possessed by agents at the end of each run. Again, we compare the results of experiments 1 and 2. For each of the corresponding match values, the average and standard deviation of the agent capital was calculated for all 289 target strings used. As the number of characters in each word was greater than 15, it can be seen that there are a number of complete solutions shown in the lower right hand part of the plots. A comparison of the two plots suggest that the diverse population has overcome, to some extent, the disadvantage of higher prices during trading, by eliminating some of the agents responsible for driving those higher prices.

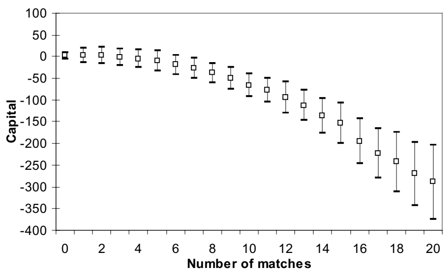


(a)

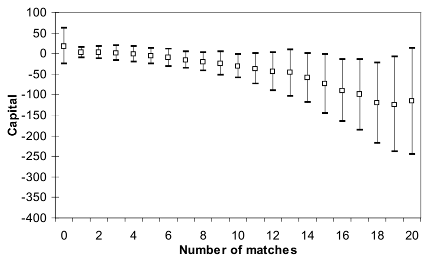


(b)

Fig. 3. The distribution of matches after 5,000 iterations: (a) Experiment 1 – using no termination; (b) Experiment 2 – using a 50% termination threshold. The average frequency of agents having 1 match is equal to 296, but the graph is shown with the same scale as (a) for purposes of comparison.



(a)



(b)

Fig. 4. Average capital owned by agents after 5000 iterations against number of matches: (a) Experiment 1 – using no termination; (b) Experiment 2 – using a 50% termination threshold.

6 Discussion and Conclusion

In this study, we have proposed a novel market-based cooperative problem solving mechanism. This incremental approach based on bottom-up team or coalition formation describes one possible mechanism for autonomous agents to coordinating their decisions without assuming *a priori* cooperation.

At the beginning of a run, agents were initialized with a random letter (resource). This particular resource may be attractive to other agents. As the model is iterated, other agents in the population may bid to gain control of the resource (resulting string fragments) in an attempt to satisfy the global objective. In this instance, the trading metaphor represents an effective communicating strategy, allowing agents to form coalitions using a bottom up methodology (Fig 2). Here, a coalition provides a framework for solving the given problem, which could not be solved by one agent working alone.

The preliminary results presented in this paper are very encouraging. The plots in Fig 3 illustrate that the agent populations is able to solve a given problem. In fact, a number of different successful coalitions have emerged, for the each of the target strings used in the simulations. The use of a notional currency preserves links between agents that otherwise yield their stake in the solution to the control of the buying agent. It is interesting to note that when some of the agents are removed from the population and replaced with new agents, an improvement in the number of matches found was noted. Fig 4 provides further empirical evidence supporting this notion based on the capital invested.

Coalition formation may be thought of generically as the process of devising a team of agents to work on a specific goal. In our model, agents continually interact with other agents and have to adapt to their environment. The repeated transactions between agents facilitate the developed of links (or trade networks). Our model is characterized by the nonlinear credit-assignment or utility function associated with the agent parameters. These parameters define the extent to which agents compete or cooperate. Coordination via this type of market mechanism is well suited for situations in which: (a) resources can be described easily or are commoditized, and (b) there are several agents offering the same (type) of resources and several agents that need them.

The cooperative problem solving model investigated in this study has focused on explicit subsystem interactions. As such, there are similarities between this work and the aggregation mechanisms inherent in Potter and De Jong's [15][16] cooperative coevolution model and the idea of combining together partial solutions into more complete solutions via sexual recombination (for example, the building-block hypothesis, Holland [3][20]). The simulation experiments described clearly illustrate two important characteristics of emergent properties in complex systems: (a) there must be a sufficiently large number of agents for the model to be effective, and (b) the model must include explicit self-reinforcing mechanisms.

This work also raises a number of questions in relation to the formation of coalitions or modules in complex systems. And, in particular how do these coalitions self-organize into hierarchies? Although the model described here is highly idealized, the underlying protocol may provide some insights into the characteristic interactions, which facilitates the transition from lower-level entities into new higher-level functional entities. Traditional practices in multi-agent systems rely on preprogrammed interaction patterns, preventing adaptation to unexpected environmental changes. A market-based bottom-up protocol may offer an alternative means for self-assembled coalition/hierarchies to emerge.

Acknowledgements. This work was partially supported by a Communities of Scholars Award from the Faculty of Science and Agriculture, Charles Sturt University. We thank Leighton Weymouth for programming support.

References

1. Wooldridge, M.: *An Introduction to MultiAgent System*s. John Wiley & Sons. UK. 2002.
2. Rosenschein, J.S. and Zlotkin, G. : *Rules of Encounters: Designing Conventions for Automated Negotiation among Computers*. MIT Press. Cambridge, MA. 1998.
3. Holland, J. H. : *Emergence: From chaos to order*. Reading, MA: Addison-Wesley. 1998.
4. Axelrod, R. : *The Evolution of Cooperation*. Basic Books, New York. 1984.
5. Huberman, B.A. : The performance of cooperative processes. In Emergent Computation – Special Issues. *Physics D*. 1991.
6. Wellman, M.P., Walsh, W.E., Wurman, P.R. and Mackie-Mason, J.K. : Auction protocols for decentralized scheduling. *Games and Economic Behaviour*. 35(1/2):271-303. 2001.
7. Tambe, M. : Towards flexible teamwork. *Journal of Artificial Intelligence Research*. 7:83-124. 1997.
8. Jennings, N. : Controlling cooperative problem solving in industrial multi-agent systems using joint interactions. *Artificial Intelligence Journa*l. 75(2):1-46. 1995.
9. Shehory, O. and Kraus, S. : Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. In *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96)* AAAI Press / MIT Press. pp 330-337. 1996.
10. Kraus, S., Shehory, O. and Taase, G. : Coalition Formation with Uncertain Heterogeneous Information. In J. S, Rosenschein et al., (eds). *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM Press. pp 1-8. 2003.
11. Sims, M. Goldman, C.V. and Lesser, V. : Self-Organization through Bottom-Up Coalition Formation. In the *Proceedings of AAMAS'03*. ACM Press. pp 867-874. 2003.
12. Sandholm, T. and Lesser, V .: Coalition formation among bounded rational agents. *Artificial Intelligence Journal.* 94(1-2):99-137. 1997.
13. Wooldridge, M. and Jennings, N.R. : The Cooperative Problem Solving Process. *Journal of Logic & Computation*. 9(4):563-592. 1999.
14. Wellman, M .: Market Oriented Programming: Some Early Lessons, In S.H.Clearwater (ed). *Market-Based Control a Paradigm for Distributed Resource Allocation*. World Scientific Press. 1996.
15. Potter, M. and De Jong, K .: A Cooperative Coevolutionary Approach to Function Optimization. In *Parallel Problem Solving from Nature Conference PPSN III*. pp 249-257. Springer-Verlag. 1994.
16. Potter, M. and De Jong, K. : Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evolutionary Computation* 8(1): 1-29. 2000.
17. Watson, R.A., and Pollack, J.B. : Symbiotic Combination as an Alternative to Sexual Recombination in Genetic Algorithms. *Parallel Problem Solving from Nature -- PPSN VI*, pp. 425-434. Springer. 2000.

18. Daida, J.M., Gasso, C.S., Stanhope, S.A. and Ross, S.J. : Symbioticism and Complex Adaptive Systems I: Implications of Having Symbiosis Occur in Nature. In *Evolutionary Programming V*. MIT Press. 1996.
19. Smith, R.G. : The contract net protocol. *IEEE Transactions on Computers*. C29(12). 1980.
20. Holland, J. : *Adaptation in Natural and Artificial Systems*. University of Michigan Press. 1975.