

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Utility Driven Coalition Formation Among Constrained Agents

¹Habiba Belleili, ²Maroua Bouzid and ¹Mokhtar Sellami

¹Laboratory of Computer Sciences, Badji Mokhtar University BP 12, Annaba Algeria

²GREYC-CNRS BD Marechal Juin, BP 5186, 14032 Caen Cedex

Abstract: This study tackles the problem of the coordination problem among constrained agents. Agents are constrained in different ways. First agents operate in a dynamic environment making difficult to plan and predict the agent's future workload; second they cannot decide about their actions without interacting with other agents; third they operate with resources limitation. Resources limitation stem from agents having multiple tasks to perform and having bounded resources in which to perform them. Finally they are hierarchical: each agent uses the result of the other to improve gradually, with the local technique, the response quality of the task. This study presents a polynomial coordination algorithm for coalition formation based on a utility function so that to respond with a best quality possible in delay (bound optimally).

Key words: Coalition formation, coordination, progressive reasoning, resource bounded agents, utility driven decision

INTRODUCTION

Traditional AI systems work off-line. Their input is a complete problem description and their output-acquired after some unknown, long time delay, is a complete answer. As problem solving systems scale up and longer execution times become common, many problem solving tasks become real-time (Garvey and Lesser, 1994), i.e., considerable changes in the real world take place during an agent deliberation. The value of an agent's action in the real world may decrease with time. New strategies are needed to find a trade-off between the solution quality and the computational resources rather than to compute an optimal solution. Resources bounded reasoning systems have been introduced to deal with such strategies. The possibility of dynamically adapting the depth of the treatment to the available time was largely studied by the community of the artificial intelligence. These efforts led to the development of a variety of techniques such as: anytime algorithms (Zilberstein and Russel, 1996), design to time (Garvey and Lesser, 1993), flexible calculation (Horvitz, 1988), imprecise calculation (Liu *et al.*, 1991) and the progressive reasoning (Mouaddib and Zilberstein, 1997).

All these approaches share the same objective which is to increase as better as possible response qualities in the time windows specified by problems in entry. It is said to be a utility driven decision making. Several works using a utility function for the decision-making were

proposed. In particular, we quote (Azulay-Schwartz and Kraus, 2002; Schwartz and Kraus, 1997) for the allocation of the data in a multi-agents environment (Mouaddib 1999; 2004a) for the coordination of progressive agents plans and (Mouaddib, 1998; 2004b) for the coordination of plans by the collection of temporal relations between them.

Present study is based on progressive processing approach which has as distinctive characteristic, the use of multi level deliberation hierarchy to transform an approximate solution to a more precise one.

Assumptions on which the approach is based correspond to a set of independent tasks each one with a deadline. Tasks are ordered by Earliest Deadline First (EDF). Tasks can be treated at various levels of details using a hierarchy of processing levels. Each processing level has probabilistic distribution of duration and a discrete quality outcome.

The aim is to return a schedule of tasks which meets respective tasks deadlines. An Incremental Scheduler (IS) was proposed (Mouaddib and Zilberstein, 1997) for this purpose. Unlike classical scheduling systems, the incremental scheduler schedules tasks levels as elementary component. Incremental scheduler presents an incremental insertion of reasoning levels starting by inserting first levels (mandatory level) of each task corresponding to the minimal schedule (fast approximate schedule) and afterwards inserts optional levels of tasks incrementally. The global utility obtained by incremental

scheduler depends on the number of levels inserted by IS without task deadline violation. The first attention of incremental scheduler is to increase as better as possible the global utility.

Our proposition rises from the following observation the incremental scheduler is centralized and the execution of tasks is sequential (task by task). This leads to increase tasks latencies and hence to decrease the global utility. The question is how can we minimize tasks latencies in order to increase as possible the global utility? This can be done by agentifying reasoning levels of progressive processing units to allow parallel execution of levels. Each agent of level i will have in charge to schedule and execute levels i of tasks with a local technique. The advantage of the agentification is the processing of more than one task at a time. For example, when the agent of the first level executes the first level of one task, it can start with the next task. The aim in the agentification is to schedule and execute more levels by task in order to obtain more performance overall by increasing the global utility of tasks outcome regarding the one obtained with the centralized Incremental Scheduler.

In order to execute a task, agents have to coalesce by taking into account their local constraints. The resulting coalition will include all agents or less agents depending on agents actual constrains. To coalesce agents need a coordination process. The coordination is necessary when no agent has sufficient resources and information to solve a problem on one hand and also to have consistent local plans on the other hand. For these reasons agents require mutual temporal information so that they can plan downstream from interaction.

In this study, we present a new distributed algorithm for coalition formation among hierarchical agents for the distributed resolution of temporal constrained problems. This algorithm seeks to minimize the cost of the coalition while maximizing the utility of the answer. We particularly study the coordination algorithm in term of complexity and time consumption and we show the gain obtained in term of global utility by the agentification we propose. This approach is suited to problems that require the result to be expressed at various levels of details such Hierarchical Planning (Knoblock, 1994) incremental Diagnosis (Hayes-Roth, 1990), mobile Robot navigation (Zilberstein, 1996) real-time data transmission (Millan Lopez *et al.*, 1994).

FORMAL FRAMEWORK

Agents system we deal with, consists of a group of hierarchical agents, each agent represents a level, which

represents a level of details of task solution. The system works as a server, where tasks arrive with an average rate asynchronously. Let $\Gamma = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ be the set of agents we deal with. We assume that the information necessary for a task is produced by the first level agent. The first level agent is hence the mandatory agent for the treatment of the task. In the following, we briefly present some general notations and concepts definitions.

Agent neighbourhood: Each agent α_i (except for agent of the first level $i = 1$ and the agent of the last level $i = n$) has two direct neighbours by default: the low level neighbour α_{i-1} and the high level neighbour α_{i+1} .

Indirect neighbour agents of low level of agent α_i are $\alpha_k (1 = k < i - 1)$. Indirect agents of high level of agent α_i are $\alpha_k (i + 1 < k = n)$.

For the hierarchical architecture of our agents system the agent of the first level α_1 has no neighbour of low level and the agent of the last level α_n has no neighbour of high level.

Progressive processing: For a task treatment, agents must form a coalition which is a temporary association between agents in order to carry out a task. As soon as a coalition is formed for a specific task W_j , each agent α_i in the coalition has a unique low level neighbour agent from which it gets an entry and a unique high level neighbour agent to which it communicates results. We note Neighbourhigh (α_i, W_j) the neighbour of high level of agent α_i for the task W_j and Neighbourlow (α_i, W_j) the neighbour of low level of agent α_i for the task W_j .

Local technique duration: Each agent maintains a probabilistic distribution of its local technique which are dynamically determined and updated by statistical analysis on the behaviour of the technique.

METRICS FOR AGENT DECISION MAKING

We are situated in a problem with temporal constraints and for reasons of workload which can compromise task deadline some agents cannot be included in the treatment of a task. The purpose of our proposal is to answer the following questions: Which are the agents that must opt out of the treatment of the new arrival task? or in other words, which are the agents that compromise task deadline? If all agents cannot be included in the treatment of a task, which are those of minimal utility? To answer these questions, we need to define a function named

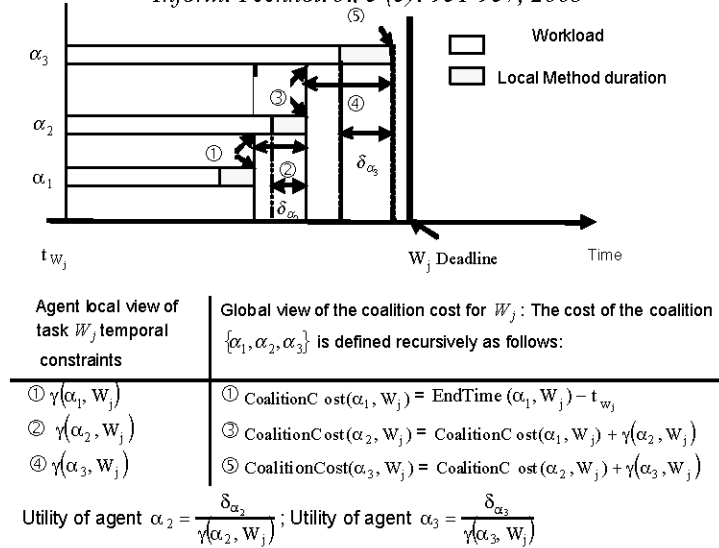


Fig. 1: Coalition cost and time dependent utility function

coalition cost which expects task deadline violation and a function which returns the utility of an agent for the treatment of a task.

Coalition cost: The coalition cost is represented by the estimated computational resources to answer a task. It depends on the cumulated costs of the previous coalitions. A coalition cost for a given task must meet task deadline.

Coalition cost for a task W_j is perceived by an agent α_i (maintained in the coalition for W_j), as local temporal information for the task W_j . These information are the expected ready time at which the low level neighbour of agent α_i for the task W_j communicates the entry (ReadyTime (α_i, W_j)) and the expected end time at which agent α_i completes its treatment on the task W_j (EndTime (α_i, W_j)). The time interval spent by a task W_j at a level of an agent α_i , returned by the function $\gamma(\alpha_i, W_j)$ (Fig. 1), contributes to estimate the cost of the coalition as a global metrics which is defined recursively as follows:

$$\begin{aligned} \text{CoalitionCosts}(\alpha_1, W_j) &= \text{EndTime}(\alpha_1, W_j) - t_{w_j} & (1) \\ \text{CoalitionCosts}(\alpha_i, W_j) &= \text{CoalitionCost}(\alpha_{i-1}, W_j) + \gamma(\alpha_i, W_j) & (2) \end{aligned}$$

Where t_{w_j} is the arrival time of task W_j . Figure 1 shows the local and global view of the coalition cost.

Time dependant utility function: We define a time dependant utility function which permits the statement of a preference order among agents for a specific task. This utility function is useful when a conflict is expected. It represents the contribution of an agent α_i to the coalition cost.

The utility of an agent α_i to a task W_j noted $g_{\alpha_i}(W_j, t)$, represents the utility of an agent α_i to respond to a task W_j , at a time t . It is defined as follow:

$$\frac{\delta_{\alpha_i}}{\gamma(\alpha_i, W_j)} \quad (3)$$

Where δ_{α_i} is the expected duration of the local method of agent α_i .

The utility function of an agent for a task W_j is not an independent function but it depends closely on temporal information of its current low level neighbour for the task W_j which are transmitted by corresponding messages. The Fig. 1 shows the dependency of the utility function in terms of temporal constraints of its current low level neighbour for the current coordination.

AGENTS INTERACTION PROTOCOL

Agents communicate via a communication protocol, which is based on two primitives:

Send/Receive(Sender, Receiver, <Message>).

There are four types of messages as follows:

Invitation-to-participate message: This type of message is an invitation of the sender to the receiver to look at the possibility to participate for the treatment of the task W_j . This message vehicles temporal information the expected time the sender delivers its result for the task W_j which are the cost of the coalition until the sender the value of the minimal utility, U_{min} , until the sender, the agent which owns the minimal utility arg (U_{min}) and task identification W_j . The agent which receives this message,

deduces from temporal information of the sender its proper temporal information such the expected ready time of the task at its level, which corresponds to the expected end time of the sender and also deduces its utility.

Invitation-to-Opt-out message: This type of message is an invitation of the sender to the receiver to opt out of the coordination because of its minimal utility. The parameter of this message is task identification.

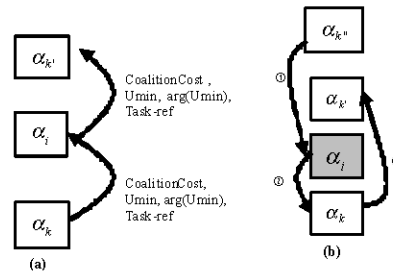
NewNeighbour message: This type of message is an updating message which permits to the receiver to know its new high level neighbour after the sender has opted out. Parameters of this message are: new neighbour identification and task identification.

NoNeighbourUp message: Only the agent of the last level sends this type of message when it must opt out due to its minimal utility. The parameter of this message is task identification. The receiver of this message deduces that it has no neighbour of high level for the treatment of the task referenced in the message and that it is the last agent in the hierarchy for the referenced task.

UTILITY DRIVEN COALITION FORMATION

On arrival of a task, agents will enter in communication to form a coalition which maximizes the utility of the response to the new task. This will be done by identifying among all agents of the set Gamma those which will take part in task response improvement while respecting its deadline. In this step, agents try to construct together a coalition so that the expected cost (corresponding to available computational resources) meets task deadline. Agents reason about available computational resources using worst case performance of their local techniques. Recall that each agent maintains a distribution of probability of its local technique obtained by statistical analysis on the behaviour of the techniques. We distinguish two main situations: the first is the progressive coalition formation; the second concerns the detection of task deadline violation. Hence, one agent must opt out. We adopt a utility driven selection, where agent with the more little utility is invited to opt out of the current coordination and hence from the treatment of the task. We address in the following these two situations.

Progressive coalition formation: When a new task W_j asks for service, agent of the first level initiates the coalition formation process. It estimates its contribution with respect to its workload corresponding to resources



(a) **Before task deadline violation:** Update the expected CoalitionCost and the value of the Minimal Utility and its Owner $\arg(Umin)$
 (b) **Expectation of task deadline violation at a level of α_k , with $\arg(Umin)=\alpha_k$**
 ① Receive an invitation to opt out message
 ② Send to low level neighbour α_k the identity of its new high level neighbour α_k
 ③ The coalition formation is resumed from the agent α_k

Fig. 2: Coalition formation principal

consumption by previous coalitions and estimates the cost of the coalition it causes for W_j using recursive formulation in (1). This information will be communicated to its neighbour of high level by default using an invitation to participate message:

Invitation-to- participate (Sender: α_1 ,
 Receiver: α_2 ,
 Temporal information: CoalitionCost (α_1, W_j)
 Umin: ∞ ,
 Arg (Unim): α_1 ,
 Task identification: W_j);

The utility of agent α_1 is always equal to ∞ for its mandatory role.

Each time an agent of the intermediate level α_i receiving an invitation to participate message from an agent of low-level for a task W_j , starts by deducing from temporal information transmitted through the message, the cost of the coalition it causes CoalitionCost (α_i, W_j) using the Eq. (2) and its utility using (3) for the task W_j , compares its utility to the one received and updates the value of the minimal utility and its owner, if necessary. If CoalitionCost (α_i, W_j) doesn't violate task deadline, it will save locally all expected local temporal information and will send them to the next agent in the hierarchy (if it exists). This process is iterated until α_n agent and the coalition is progressively formed Fig. 2a.

Expectation of task deadline violation: When at a level of an agent α_i a violation of task deadline is expected (the expected coalition cost violates the deadline of the task W_j), agent α_i has to identify among agents of low level the

agent with minimal utility for the treatment of the new task. We recall that this information is available in the invitation-to-participate message. We distinguish two cases:

- The agent with minimal utility is the agent α_i itself; it consequently must opt out of the coalition. Before opting out, agent α_i cooperates by linking its current neighbour of low-level with its current neighbour of high level, if it exists, by sending a NewNeighbour message to its current neighbour of low level with the identity of its new neighbour of high level. In the case where no neighbour of high level of an opting out agent α_i exists, agent α_i sends a NoNeighbourUp message to its neighbour of low level.
- When the agent with minimal utility is at the bottom of the hierarchy, agent α_i sends an invitation to opt out message to the owner of the minimal utility (Fig. 2 (1)).

When informed (by an invitation-to-opt-out message), the agent with minimal utility $arg(U_{min})$ cooperates, for coherent behaviour of the coordination, by linking its current low-level neighbour for the referenced task with its current neighbour of high level by sending an updating message (Fig. 2b (2)). Only at this time, the agent with minimal utility $arg(U_{min})$ opts out.

The agent receiving the updating about its new neighbour of high level for the referenced task has to resume the coalition formation process (Fig. 2b (3)) by sending its temporal constraints, saved for such event, to its new high level neighbour agent for the referenced task. The latter, must re-estimate its temporal constraints with respect to those received: CoalitionCost, minimal utility U_{min} the owner of the minimal utility $arg(U_{min})$ and the identity of low level neighbour (the sender).

At the end of the coalition formation, each agent maintained for the treatment of the new task, must know its neighbours (high and low levels).

ANALYSIS

We analyze our proposition in terms of complexity and experiments evaluation.

Complexity: Our coordination algorithm presents backtracking, we can have backtracking when a violation of the task deadline is expected. The worst case backtracking is the one where the evaluation of the coalition cost has progressed until the last agent, at a level of which a task deadline violation is expected and the agent with the minimal utility is situated at the bottom

of the hierarchy. Example: for $n = 5$, the evaluation of coalition cost is done from agent 1 to agent 5, at the last level agent a task deadline violation is expected, the agent with the minimal utility is situated at the bottom of the hierarchy which corresponds to agent of level 2 (agent of the first level is the mandatory agent). The complexity of the worst case backtracking for n agents is $O(n)$. For $(n-1)$ possible backtrackings the complexity in the worst case is $O(n^2)$.

Experimental results: To evaluate our approach which is the agentification of reasoning level of progressive reasoning, we compare it to the original work (i.e., progressive reasoning) where an incremental scheduler is proposed. This one is interruptible because it has always a response at hand (first levels of tasks are first scheduled) and depending on the tightness of tasks deadline other levels are inserted. More levels are inserted, higher is the global utility. We have compared our distributed approach to the centralized one by varying environmental parameters which are the tightness of deadlines and the number of tasks. Without surprising, the approach we propose allows insertion of more levels represented by agents, which permits a better global quality particularly in case of

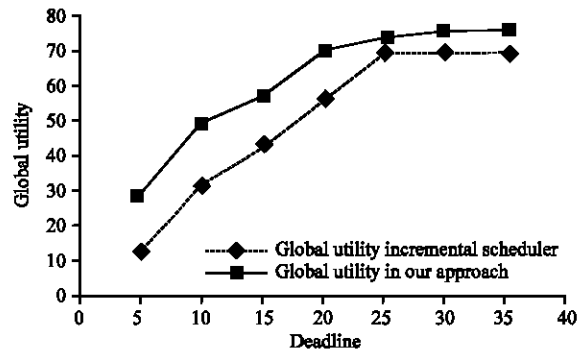


Fig. 3: Global utility and deadline tightness

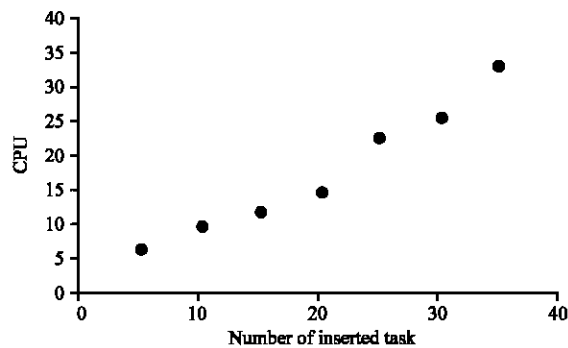


Fig. 4: Overhead of the coordination in terms of inserted tasks

tightness deadlines. This is because of concurrent execution of levels (agent). The global utility becomes near to the one obtained by the incremental scheduler when the deadline is relatively large (Fig. 3).

We have experimented (Fig. 4) our coordination algorithm in term of number of tasks and CPU consumption. The results correspond to the ones obtained theoretically. The overhead of the coordination is reasonable and doesn't affect the performance overall corresponding to a better global utility which is the first goal of any resource bounded approach.

DISCUSSION

In this study we have proposed a completely distributed coordination algorithm for coalition formation among hierarchical resource bounded agents. Classical coalition formation algorithms search the coalition among all possible coalitions (Sandholm and Lesser, 1995; Shehory and Kraus, 1998). Indeed, the number of possible coalition for n agents is 2^n coalitions which lead to an exponential complexity. This solution cannot be applied in time-bounded context. The coordination algorithm we present constructs a coalition, progressively with a polynomial complexity in the worst case. The resulting coalition meets task deadline while maximizing the quality outcome,

The approach we propose makes trade-off between the quality outcome and the available computational resource. As a matter of fact, depending on the available time different coalitions can be formed ($\alpha_1, \alpha_3, \alpha_4, \alpha_5$ or $\alpha_1, \alpha_2, \alpha_4$ or $\alpha_1, \alpha_3, \alpha_4, \dots$) the quality outcome is measured in terms of the number of agents in a coalition. When all agents cannot be included in task treatment because of resource limitation, a utility driven coalition formation is made by discarding agents with the minimal utility. The price is a backtracking which is materialized by the resume of the estimation process of coalition cost. We have proposed a utility function which allows the statement of preference order among agents and the agent with the minimal utility is discarded.

The agentification of reasoning levels with the coordination algorithm we propose, have allowed to reach the original objectives by reducing tasks latencies and increasing the global utility compared with the one obtained with the Incremental Scheduler. Experimental results confirm the gain obtained by the approach we propose in term of global utility particularly in the case of deadline tightness.

In early work (Belleili *et al.*, 2004), we have proposed a faster algorithm, for coalition formation. This algorithm discards the agent at the level of which the expected

coalition cost violates task deadline. This algorithm even faster because there is no backtracking, the resulting coalition is not utility driven and hence doesn't necessary maximize the utility of task response.

In this study, we don't address how agent reacts when the expected End Time of a local technique is earlier than expected. Recall that agents plan using the worst-case performance of their local techniques, this leads not to use resource efficiently. To deal with unexpected situation, we need additions to the coordination algorithm and monitoring of method performance. Mainly, when an agent opts out of a specific coalition, because of its minimal utility and when it detects that it has over estimated its response time, we want to give it the possibility to repair by asking to join a coalition with its new temporal constraints.

We have proposed in (Belleili *et al.*, 2005) a multistage coordination algorithm which is an extension of the approach proposed in this study. The idea is to provide agents with alternative methods, which is another resource bounded reasoning approach and also we give each agent a meta-level controller which allows it to reason about the consumption of time by the coordination activity. The extension is under experiments.

REFERENCES

- Azulay-Schwartz, R. and S. Kraus, 2002. Negotiation on data allocation in multi-agent environments. *Autonomous Agents and Multi-agent Syst. J.*, 5: 123-172.
- Belleili, H., M. Bouzid and M. Sellami, 2004. Anytime negotiation between agents with alternative methods. In *IADIS International Conference on Applied Computing*, pp: 73-80.
- Belleili, H., M. Bouzid and M. Sellami, 2005. Cooperative scheduling among time-bounded agents. *17th International Conference on Tools with Artificial Intelligence*, pp: 573-577
- Garvey, A. and V. Lesser, 1993. Design-to-time real-time scheduling. *IEEE Transaction on Systems, Man and Cybernetics*, 23: 1491-1502.
- Hayes-Roth, B., 1990. Architectural Foundation for real-time performance in intelligent agents. *J. Real-time Sys.*, Vol. 2.
- Horvitz, E., 1988. Reasoning under varying and uncertain resource constraints. *7th National Conference on Artificial Intelligence*, pp: 111-116.
- Knoblock, C.A., 1994. Automatically generating abstractions for planning. *Artificial Intelligence*, 68: 243-302

- Liu, J., K. Lin, W. Shih, A. Yu, J. Chung and W. Zao, 1991. Algorithms for scheduling imprecise computations. *IEEE Transactions on Computers*, 24: 58-68.
- Millan-Lopez, V., W. Feng and J.W.S. Liu, 1994. Using imprecise computation technique for congestion control on real-time traffic switching element. *International Conference on Parallel and Distributed Systems*.
- Mouaddib, A.I. and S. Zilberstein, 1997. Handling duration uncertainty in meta-level control of progressive reasoning. *15th International Joint Conference on Artificial Intelligence*, pp: 1201-1206.
- Mouaddib, A.I., 1998. Multistage negotiation for distributed scheduling of resource-bounded agents. In *AAAI Spring Symposium on Satisfying Models*, pp: 54-59.
- Mouaddib, A.I., 1999. Anytime coordination for progressive planning agents. *AAAI*, pp: 564-569.
- Mouaddib, A.I., 2004a. Incremental coordination for time-dependent bounded agent. *Intl. J. Artificial Intelligence Tools*, 13: 511-531.
- Mouaddib, A.I., 2004b. Co-operation scheduling for a resource-bounded multi-agent planning system. *J. Exp. Theo. Artif. Intel.*, 16: 57-71.
- Sandholm, T. and V. Lesser, 1995. Coalition formation among bounded rational agents. In *IJCAI*, 95: 662-669.
- Schwartz, R. and S. Kraus, 1997. Negotiation on data allocation in multi-agents environments. In *AAAI*, pp: 29-35.
- Shehory, O. and S. Kraus, 1998. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101: 165-200.
- Zilberstein, S. and S. Russel, 1996. Optimal composition of real time systems. *Artificial Intelligence*, 82: 181-213.
- Zilberstein, S., 1996. Using anytime algorithms in intelligent systems. *AI Magazine*, 17: 73-83.