Formation of Virtual Organizations Through Negotiation

Mark Hoogendoorn[1] and Catholijn M. Jonker[2]

[1] Vrije Universiteit Amsterdam,
Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
mhoogen@cs.vu.nl
[2] Radboud University Nijmegen,
Nijmegen Institute of Cognition and Information
Montessorilaan 3, 6525 HR Nijmegen, The Netherlands
C.Jonker@nici.ru.nl

Abstract. In this paper negotiation is presented as a solution to the formation of virtual organization in domains with many parties having (partially) unknown constraints and profiles and in which the environment is dynamic by nature. The solution presented is based on the MAGNET negotiation system, for which an extension is presented, that allows for last minute changes and failure management. An efficient algorithm is presented for supplier agents, incorporating preferences, and other constraints related to existing individual plans). Combining the algorithms for supplier agents, with a simple customer agent specification, and the ability to iterate the bidding, MAGNET is extended to deal with domains as described above. A case study in logistics using real data from a logistics company shows the validity of the approach.

1 Introduction

Virtual organizations have been defined as organizations where "complementary resources existing in a number of cooperating companies are left in place, but are integrated to support a particular product effort for as long as it is viable to do so" [7]. Nowadays, companies tend to outsource many non-core operations to upstream and downstream partner firms whose capabilities complement their own [9]. The relationship between such firms precisely complies to the definition of a virtual organization, making it an interesting type of organization to investigate given the current trends in organization theory.

Existence of a virtual organization can be long term or short term, where in the latter case the organization might only be formed to perform a few tasks. Especially for the cases where only a small number of tasks is involved in formation of a virtual organization, the overhead of the formation itself might be relatively large, possibly even causing more time than the task itself. One crucial aspect that for instance needs to be addressed in the formation process is what agents to allocate to what tasks. In order to cope with this problem, techniques from AI are being used to reduce the effort accompanying formation of a virtual organization.

K. Fischer et al. (Eds.): MATES 2006, LNAI 4196, pp. 135 – 146, 2006.

This paper presents the application of one AI technique, namely automated negotiation between agents, to formation of virtual organizations. More in particular, the paper presents a system which enables automated allocation of agents to particular tasks that need to be performed within the virtual organization. The system tries to find a suitable allocation of tasks from two perspectives: (1) that of the agent looking for an agent to perform the task, and (2) that of the agent who can perform the task. Since both can have different, possibly partially conflicting interests, negotiation is most suitable to get to a solution for both parties. Besides the initial formation, the system also has facilities to cope with failure of agents to perform their allotted tasks and to redistribute tasks.

Section 2 presents MAGNET as the negotiation platform the supplier and consumer agents can use to find a solution to their needs. The techniques and extensions needed to be able to use the MAGNET for the dynamic formation of virtual organizations are presented in Section 3. Special attention is paid to obtain robustness with respect to failures in task performance and changes in the environment warranting the change of existing virtual organizations and the formation of new virtual organizations capable to cope with the situation at hand. The system was tested using real data from a logistic company. The test results are presented in Section 4. Section 5 discusses alternative approaches in literature and presents the conclusions.

2 The MAGNET System

This Section describes the negotiation system used as a basis for the development of the system supporting the formation of virtual organizations. The negotiation system used is the MAGNET (for Multi-AGent NEgotiation Testbed) system [4]. In [1] the MAGNET system is described as follows: the MAGNET architecture provides a framework for secure and reliable commerce among self-interested agents. What makes MAGNET particularly suitable is its ability to support negotiation of contracts for tasks that have temporal and precedence constraints [4]. MAGNET shifts much of the burden of market exploration, auction handling, and preliminary decision analysis from human decision-makers to a network of heterogeneous agents. Two types of agent are distinguished within such a network: The *supplier* agent and the *customer* agent. The main interactions between the two agent types are as follows:

- A customer agent issues a *Request for Quotes* (RFQ) which specifies tasks, their precedence relations, and a time line for the bidding process. For each task, a time window is specified giving the earliest time the task can start and the latest time the task can end.
- Supplier agents submit bids. A bid includes a set of tasks, a price, a portion of the price to be paid as a non-refundable deposit, and estimated duration and time window data that reflect supplier resource availability and constrain the customer's scheduling process.

- The customer agent decides which bids to accept. Each task needs to be mapped to exactly one bid (i.e. no free disposal [11]), and the constraints of all awarded bids must be satisfied in the final work schedule. In MAGNET the customer can chose from a collection of winner-determination algorithms (A*, IDA*, simulated annealing, and integer programming).

The customer agent awards bids and specifies the work schedule.

3 Formation of a Virtual Organization

An overview of the activities accompanying the formation of a virtual organization supported by the system introduced in this paper is presented in this Section. Note that for evaluation and communication concerning the negotiation the MAGNET system can be used whereas more specific implementations for the *customer* and *supplier* agent are needed for specific domains such as the formation of virtual organizations.

3.1 High-Level System Overview

A high-level activity diagram of the system is shown in Figure 1. At the starting point the tasks to be fulfilled by the virtual organization come in, which are bundled in an RFQ and sent via the MAGNET system. The RFQ is sent to all *supplier* agents that might want to participate in the virtual organization. These *supplier* agents bid on the tasks they are able to perform and prefer and send a bid including these tasks back via the MAGNET system. After receiving all the bids, the MAGNET system evaluates these and selects the best set of bids possible. In case this set does not fully cover the tasks, an RFQ is sent again. For the bids that are in the set of optimal bids, an award is sent. The *supplier* agent that receives such a reward takes place in the virtual organization and starts executing the tasks, possibly reporting trouble requiring sending another RFQ for the task. Finally, after all tasks have been performed, the virtual organization is terminated.

3.2 Customer Agent

The *customer* part of the system mainly includes the formation of Request for Quotes (RFQs), the sending of awards for bids, and reassignment of tasks which are not properly performed. Tasks in the system include the following elements: intake time, early start time, late start time, deadline, and a task description, including details on the task and constraints. After an RFQ is sent, the *customer* eventually gets a set of bids to be awarded from the MAGNET system. In case there is no bid for a particular task, a new RFQ is sent concerning the particular task. After a task is assigned by means of awarding a bid, the *supplier* agent is placed in the virtual organization and starts to perform the task, which can result in an error report. In case such a report is received, a new RFQ with the task is sent to ensure that the task is eventually performed.

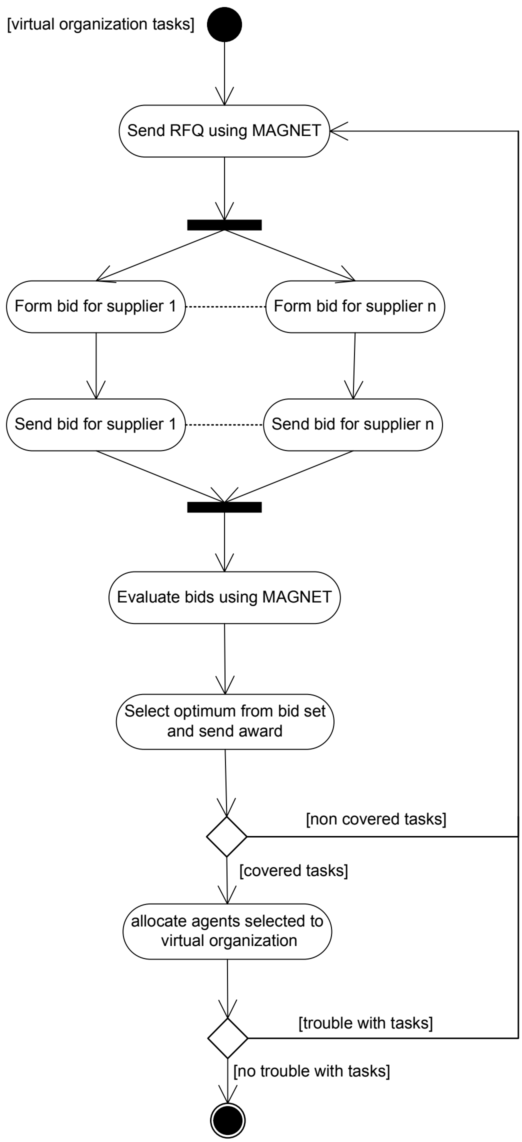

Fig. 1. UML Activity Diagram for the System

3.3 Supplier Agent

The *supplier* agents in the system are assumed to have one particular resource available during a certain time interval. Furthermore, a *supplier* is attributed with a certain preference for particular tasks, for example using the resource for a short time or using it in the beginning of the availability interval. In order to be able to derive which tasks a *supplier* is to bid on, this Section presents an algorithm which derives which tasks are included in the bid, determines the cost, and finally, determines the time windows to be inserted. The notation used for the algorithm is shown in Table 1.

Table 1. Language used in the pseudo code

Function	**Explanation**
first_task: RFQ → TASK	The function application provides the task with the earliest early start time in the RFQ.
next_task: RFQ x TASK → TASK	Results in the task with the first earliest start time in the RFQ later than the earliest start time of the specified task.
number_of_tasks: RFQ → INTEGER	The number of tasks in the RFQ.
earliest_start: TASK → TIME	Denoting the earliest start time for executing the task.
latest_start: TASK → TIME	The latest possible start time for executing the task.
expected_duration: TASK → DURATION	The expected duration of executing the task.
latest_finish: TASK → TIME	Denoting the latest possible finish time of the execution of the task.
preference: TASK → REAL	The preference value for the task, a value between 0 and 1.
last_task_before: SCHEDULE x TIME → TASK	The last task in the schedule with a latest finish before the specified time.
next_task: SCHEDULE x TASK → TASK	Specifying the next task in the schedule.
switch_time: TASK x TASK → DURATION	The time needed to switch from one task to another.
determine_risk: REAL → REAL	The risk factor taken, based on the preference for the task.

The algorithm is specified in pseudo code below, a current schedule s from the supplier's perspective with tasks already scheduled is assumed to be present in advance.

```
   t = first_task(RFQ)
   do{
     before   = last_task_before(s, earliest_start(t))
     after    = next_task(s, before)
     chi      = determine_risk(preference(t))
     duration = chi * (expected_duration(t) + switch_time(before, t) + switch_time(t, after))
     if (earliest_start(t) + duration ≤ latest_start(after))
       if (preference(t) > phi ||
          number_of_tasks(RFQ) == 1)){
          // Add the task to the bid and schedule, set the cost using a particular cost function
       }else{
           // Do not include the task
       }
}while(t = next_task(RFQ,t) && t != NULL)
```

As can be seen, the first task to be performed is taken out of the RFQ. Given the current schedule, the task just ending before the early start time of the current RFQ task is obtained as well as the task after that. Furthermore, based on the preference (a value between 0 and 1) for the current RFQ task, the amount of risk to be taken is determined (e.g. I like this task so much, I will be able to perform it faster than average) represented by χ. Now calculate the *expected* duration for performing the task, which includes switching from the previous task, performing the task itself, and switching to the next task in the schedule. The *assumed* duration to be used in the calculation is obtained by multiplying this with the χ factor. In case the duration added to the earliest start time for task t is before the latest start time of the next task, then the task can in principle fit within the schedule. There is however still the preference of the supplier, which is specified by means of ϕ. ϕ is the threshold for the preference value above which a task is preferred. In case a task is preferred and fits within the current schedule, add the task to the bid. Do the same in case the RFQ contains one single task. This reflects the understanding by the *supplier* that this is a task that really needs to be performed and for which it is hard to get somebody. The global result is that unpopular tasks also will be performed. Once a task is added to the bid, the cost for performing the task are added by means of a cost function. Two cost functions are used in this paper, where the first is simply the *assumed* duration for the task. The second cost function used is the *assumed* duration divided by the preference value, which means a higher price for less preferred tasks. One element not addressed in the algorithm is determination of time windows to be included in the bid, which is specified in pseudo code below.

```
if (chi ≤ 1){
    earliest_start = latest_finish(before) + chi * switch_time(before, t)
    if (earliest_start < earliest_start(t)){
        earliest_start = earliest_start(t)
    }
    latest_start = latest_finish(before) + switch_time(before, t)
    if (latest_start < earliest_start(t)){
       latest_start = earliest_start(t)
    }
}else{
    earliest_start = latest_finish(before) + switch_time(before, t)
    if (earliest_start < earliest_start(t)){
        earliest_start = earliest_start(t)
    }
    latest_start = latest_finish(before) + chi * switch_time(before, t)
    if (latest_start < earliest_start(t)){
       latest_start = earliest_start(t)
    }
}
duration = expected_duration(t) * chi
```

In case the χ value is less than or equal to 1 (i.e. the task is *assumed* to go faster than *expected*), then set the earliest start time to either the earliest start time specified for the task or, in case this is not feasible, to the latest finish time of the task before in the schedule plus the *assumed* switching time. The latest start is set to the latest finish time of the task before plus the *expected* switch time or, in case before the specified earliest start time, the earliest start time specified in the RFQ. If the value of χ is greater than 1, the earliest and latest start times are calculated just opposite from less than or equal to 1. Finally, the duration is set to the assumed duration.

After having sent a bid, a bid award is possibly received, resulting in the task actually being executed. The schedule is therefore replaced by a schedule including the tasks that have been awarded.

In the execution phase, incidents can occur that require replanning by the *customer* agent (or in similar domains, leading to the supplier agent becoming a customer agent that is seeking another supplier agent to solve his task). Three types of incidents are distinguished: (1) A simple task delay, that requires no replanning; (2) A task failure, the task needs to be performed by another *supplier*; (3) A day failure, all tasks for the day need to be re-planned.

4 Case Study

This section presents the results of a case study performed in order to validate the virtual organization formation approach presented in this paper. First, the domain in which the case study has been performed is described, thereafter the results regarding system performance are presented.

4.1 Case Study Description

In order to obtain experimental results, a choice has been made to use real company data instead of randomly generated data. Using company data has as the advantage that it can be determined how well such a system would work in a real environment instead of an artificially created one. The data has been obtained from a company within the field of logistics. This area is particularly interesting for application of the system due to the movement of several companies to so called Fourth Party Logistics (4PL), see e.g., [2]. A 4PL logistics company is an intermediate link within the chain of transporting goods, it closes contracts with large parties to arrange the logistics across the entire supply chain of the organization. 4PL companies have a limit amount, or possible even no trucks of their own (see e.g. [6]). They therefore have contracts with a number of trucking companies which they can call in case they need a truck for a particular order. The price for such a trip is negotiated over the phone. In the case study, the 4PL does not negotiate with the trucking companies through a scheduling officer, but directly with the truck drivers of that company. In this way the truck drivers get a higher responsibility for creating a revenue for the company they work for and they get the opportunity to guard their own preferences. Hence, the 4PL company is the *customer* in the system described in the previous section, and the trucks are the *supplier* agents, where a formation of a virtual organization for the transportation of certain goods is the goal of the negotiations.

The data used for the experiments concerns transportation of containers, of which only one can be carried at the same time by a truck. As a result, trucks can only perform tasks in sequential order and not in parallel. Furthermore, there are different types of containers: 20 feet and 40 feet containers, both of them can only be carried by a truck suitable for that particular type. Each of the tasks contain an intake time (around which the order to transport the container comes in, and thus the time at which an RFQ can already be sent), an early start time (when the container becomes available), a deadline (when the container needs to be delivered), and a start and end location. Precedence constraints are present as well in case a container has to be transported along several locations. The data obtained from the company mainly concerns container transports from one of the container terminals at the port of Rotterdam (there are several such terminals in the port) to a particular customer, after which the container needs to be returned to a certain location. Typically, about 20 orders are received each day, most of which require a pickup early in the morning. For the usage of the system presented in Sections 2 and 3, each truck is seen as a separate *supplier* where the resource is in this case the ability to transport a particular type of container. On average, about 10 trucks are available as *suppliers* per day. Trucks have a start location at which they are located at the beginning of the day (typically close to the port of Rotterdam), and have a start and end time (e.g. the trucks starts at 9 am and stops at 5 pm). Preferences of trucks are found in the different pickup and destination locations, the length of the trip required to perform the task, and the start and end times of the tasks. As a result of interviews with personnel from the data providing company, these preferences have been determined for each truck, based on the driver assigned to it. The real cost for performing a task is set to the travel time in minutes to perform the task (i.e. driving to the pickup location, performing the task, and returning from the destination location). Note that this can differ from the price actually put in the bid for the task.

4.2 Case Study Results

In order to evaluate the effectiveness of the system, simulation runs have been performed using the real life data from the trucking domain as described before. For this purpose, the logs of the order system of one representative week has been used. Using this data, the system is evaluated from two perspectives. First, the time needed to evaluate the bids is measured, to see whether this evaluation process itself is not a bottleneck within the virtual organization formation process. The algorithm for the supplier agents can be run in parallel, which is not the case for the customer agent. Another perspective from which the system is evaluated is to see how different cost functions and preference thresholds influence the overall satisfaction of the *supplier* agents within the system.

Algorithm Performance. First, the performance of the evaluation algorithm during the simulation runs is presented. Note that these results are specific results for the characteristics of the data. For more generic results on algorithm performance and a comparison between different algorithms, see [3]. The experiments have been run on a Sun UltraSPARC IIIi 1062 MHz CPU with 2 GB memory. Figure 2 shows the results of the IDA* algorithm used for the case study for RFQs with varying amount of tasks.

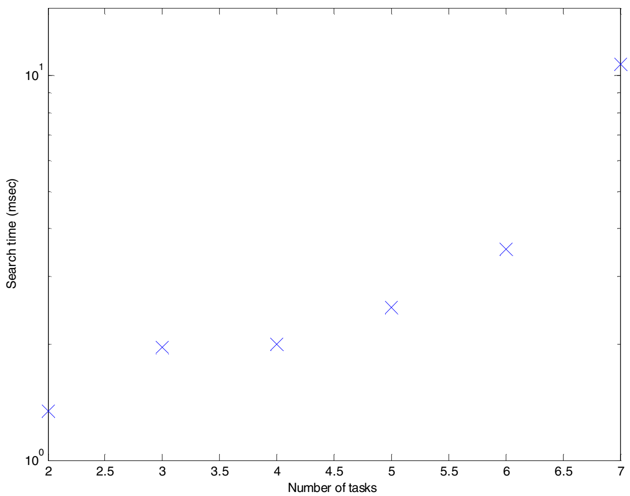

Fig. 2. IDA* search time for different number of tasks

Furthermore, Table 2 shows more detailed characteristics for the evaluation process.

Table 2. Evaluation characteristics

Number of tasks	Average number of bids	Average number of tasks in bid	Search time IDA* (msec)
2	3.59	1.00	1.35
3	6.05	1.07	1.98
4	6.00	1.00	2.00
5	7.25	1.08	2.50
6	8.00	1.25	3.54
7	7.63	1.01	10.69

As can be seen in the table, the average amount of tasks per bid is always close to one, which is due to the fact that an RFQ in the trucking domain typically specifies several tasks which need to be performed in parallel and, as already stated in the introduction of the case study, the trucks cannot execute tasks in parallel. Since only full bids can be awarded, they therefore often only bid for one task. As the graph shows, also for the RFQ's with the largest amount of tasks observed in the data (i.e. seven tasks in one RFQ) the evaluation algorithm generates a solution in just over 10 milliseconds. For a more extensive discussion on the scalability of the IDA* algorithm within the MAGNET system, see [3].

Supplier Satisfaction. Besides the evaluation time, the satisfaction of *suppliers* is another element which has been investigated. The satisfaction of the suppliers is measured in the average preference for the tasks they get awarded. Two parameters can be varied regarding this satisfaction, namely the threshold value ϕ and the function for cost to be included in the bid (i.e. assumed duration or assumed duration divided by the preference). Figure 2 shows the satisfaction of the different agents for both cost functions for varying ϕ values. Note that despite the threshold for bidding on tasks, tasks can still be bid upon in case only one task is included in the RFQ. As a result, the satisfaction can be below the threshold value set.

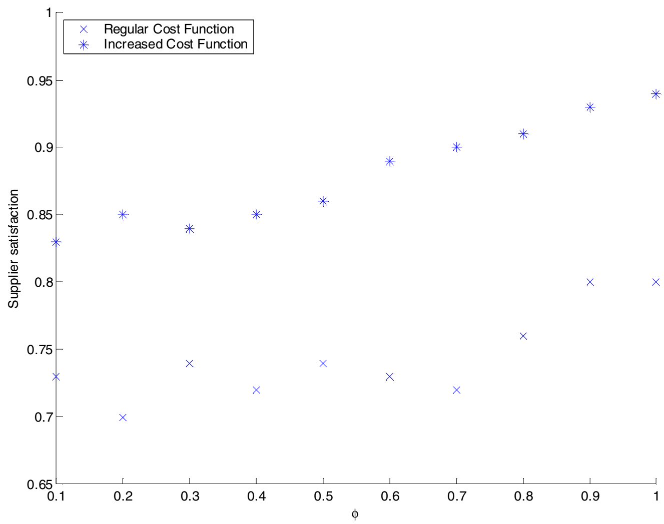

Fig. 3. Driver satisfaction for varying ϕ values

As can be seen in Figure 3, the increase of the price in case a task is not preferred is shown to be effective in the simulation runs of the case study. Having such a preference requires a less strict setting of the preference threshold ϕ for bidding on a task still obtaining a reasonable satisfaction rate. When looking at the regular price option, satisfaction is much lower once the value for ϕ decreases. An additional performance measure is of course the efficiency of the solution found, which in the trucking domain can be measured by means of the amount of effective driving (the amount of driving for a task divided by the total amount of driving). In the simulation runs, no correlation was found between the setting for the preference and the effectiveness of driving. On average, 62% of all driving was effective.

5 Discussion

This paper presents an approach for the formation of virtual organizations in highly dynamic environments which require a low overhead for the formation process of the organization. The approach allows for the formation of such an organization without the different parties needing to have knowledge about each others constraints and profiles. The approach is based upon an existing negotiation system called the

MAGNET system which is extended with specific implementations for the *supplier* and *customer* agents for the formation of virtual organizations. The implementation of the *supplier* agent incorporates preferences for tasks as well as schedules specifying the tasks to be performed. In a case study in the trucking domain, the paper shows that the evaluation algorithms incorporated in the MAGNET system scale well, requiring a minimal time for the evaluation process. Furthermore, reflecting the preference of a task in the price bid for that task in the algorithm increases the overall satisfaction of the *supplier* agents.

In the field of virtual organizations, negotiation systems have been introduced and used as well. In [8] a virtual office system is mentioned called *SmartProcurement* which is said to initiate the formation of a virtual organization by means of an electronic or human request for quotation (RFQ). Thereafter, a purchasing agent acquires a list of agents which are known vendors of the requested item and sends the RFQ to the vendors. Subsequently, the bids are evaluated and a bid is selected, informing the vendor agent upon acceptance. The approach is however more meant as a framework to support such negotiation, similar to the MAGNET system, not as a specific implementation of the agents themselves.

Besides the MAGNET system, more negotiation systems have been developed. The advantage of the MAGNET system is the market infrastructure in between the *supplier* and the *customer* agent whereas most other negotiation systems focus on direct agent to agent negotiation [12, 5] (from [1]). Based on the MAGNET system more extensive *supplier* agents have been developed [1], however these agents have not been tested with real life data. Furthermore, [1] does not focus on the formation of virtual organizations.

Team or coalition formation is another related field. Different protocols for the formation of coalitions are compared in [13]. Variations of such protocols go from local to social utility based negotiation systems. The authors show that increased social context can improve system performance. The agents are however required to share meta-level information before they allocate resources. In the trucking domain, however, agents do not want to share such meta-level information, as they might be competitors. Therefore the approach presented in [13] is not feasible in domains in which the agents represent competitors.

Different role-allocation and reallocation algorithms are compared in [10] The comparison is based on for the framework developed for the Role-based Markov Team Decision Problem. In the future the same framework could be applied to compare the approach presented in this paper with other role-allocation algorithms with respect to the corporate data for the trucking domain.

Acknowledgements

The authors would like to thank Maria Gini from the Department of Computer Science and Engineering at the University of Minnesota for the fruitful discussions. Furthermore, the authors wish to thank the anonymous reviewers for their useful comments that helped to further improve the paper.

References

1. Botman, S., Hoogendoorn, M., Bud, V., Jaiswal, A., Hawkins, S., Kryzhnyaya, Y., Pearce, J., Schoolcraft, A., Sigvartsen, E., Collins, J., and Gini, M., Design of supplier agents for an auction-based market. In: Giorgini, P., Giorgini, P. Lesperance, Y., Wagner, G., Yu, E. (eds.), Proceedings of the Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS 2002 @ AAMAS-02), July 2002.
2. Briggs, P., The hand-off: the future of outsourced logistics may be found in the latest buzzword [Fourth Party Logistics], Canadian Transportation Logistics 102(5), pp. 18, 1999.
3. Collins, J., Solving Combinatorial Auctions with Temporal Constraints in Economic Agents. PhD thesis, University of Minnesota, June 2002.
4. Collins, J., Gini, M., and Mobasher, B., Multi-agent negotiation using combinatorial auctions with precedence constraints. Technical Report 02-009, University of Minnesota, Department of Computer Science and Engineering, Minneapolis, Minnesota, February 2002.
5. Fatima, S.S. and Wooldridge, M.. Adaptive task resources allocation in multi-agent systems. In Proc. of the Fifth Int'l Conf. on Autonomous Agents, pp. 537-544, 2001.
6. Foster, T., 4PLs: The next generation of supply chain outsourcing? Logistics Management and Distribution Report 38(4), pp. 35, 1999.
7. Goldman, S., Nagel, R., Preiss, K., Agile Competitors and Virtual Organizations, Van Nostrand Reinhold, New York, 1995.
8. O'Leary, D.E., Kuokka, D., and Plant, R., Artificial Intelligence and Virtual Organizations, Communications of the ACM 40(1), pp. 52-59, 1997.
9. Miles, E.R., Snow, C.C., Mathews, J.A., Miles, G., and Coleman, H.J., Organizing in the knowledge age: Anticipating the cellular form, Academy of Management Executive 11(4), pp. 7-20, 1997.
10. Nair, R., Tambe, M., Marsella, S., Role Allocation and Reallocation in Multiagent Teams : Towards a Practical Analysis, In: Proceedings of the Second Conference on Autonomous Agent and Multi-Agent Systems (AAMAS 2003), pp. 552-559, ACM Press, 2003.
11. Nisan, N., Bidding and allocation in combinatorial auctions. In *1999*□ NWU Microeconomics Workshop, 1999.
12. Sandholm, T.W., Negotiation Among Self-Interested Computationally Limited Agents PhD thesis, Department of Computer Science, University of Massachusetts at Amherst, 1996.
13. Sims, M., Goldman, C.V., and Lesser, V., Self-Organization through Bottom-up Coalition Formation, In: Proceedings of the Second Conference on Autonomous Agent and Multi-Agent Systems (AAMAS 2003), pp. ACM Press, 2003.