Residual resources exploitation in a coopetitive MAS

Gaël Hette and **Sylvia Estivie** and **Emmanuel Adam** and **René Mandiau** [1]

Abstract. In a dynamic system, some disturbances could occur during the execution of agents task schedules. In this paper, we aim to propose a framework to solve such kind of problems by pooling residual resources to reallocate tasks during the execution of the system. In our context, the initial task schedules of agents are defined such that during some given time windows agents could perform additional tasks with their residual resources. In this paper, we study this problem of reallocation of residual resources in order to propose a solution to these disturbances.

1 Introduction

Over the last decades multi-agent paradigm has been used as a framework to solve various kinds of resource allocation and task scheduling problems in distributed systems. In order to cope with the characteristics of each problem types, different approaches have been proposed in the field of multi-agent systems. One approach consists in considering and formalizing such problems as Distributed Constraint Satisfaction Problem (*DCSP*). Another approach which forms a noticeable growing field of research, known as *Multi-agent Resource Allocation* (*MARA*), is dedicated to issues regarding allocation of resources within a system of autonomous agents [2]. A multi-agent resource allocation problem is defined as a process of distributing a number of resources (or tasks) among several agents. This distribution is conducted by a centralized or distributed allocation procedure wherein agents participate in the elaboration of the allocation by reporting their preferences over the set of resources (or tasks) they will receive. In the case of a task allocation procedure, an usual objective is to find a feasible allocation. Whereas a typical objective of resource allocation procedure is to find an optimal allocation in the sense of a measure of the social welfare arisen by the considered society of agents.

In this paper we choose to focus on the case of task allocation over a multi-agent system, where each agent disposes of non shareable resources to achieve tasks. In this context, some specific allocation problems lead to solutions where the totality of the resources are not necessary allocated. These classes of problems implie the fact that non-allocated resources (i.e. *residual resources*) are still available during the execution of such allocated system instances. In this paper we propose to use *MARA* techniques to re-allocate tasks during the execution of a defined schedule. Thus considering a failure during the execution of such schedule, some tasks become impossible to service. These tasks could be reallocated to agents disposing of *residual resources* corresponding to their achievement constraints. We aim to propose a framework to resolve such problems by pooling *residual resources* to reallocate tasks during the execution of the system. Considering an agent in need of resources, our approach is based on a model and a set of methods to find corresponding available resources within the system and then use negotiation to determine a new tasks allocation.

The paper is structured as follows. Section 2 is a detailed presentation of the problem addressed by our study and goes on by presenting a brief state of the art. Section 3 specifies the basic elements of our computational model. Section 4 details how we can apply our approach to the problem of adaptive resource management in a transportation network.

2 Preliminaries

2.1 Our problem

We consider a multi-agent system with an initial resource allocation amongst agents. This initial resource allocation is computed in order to accomplish an associated task schedule. Each tasks of this schedule have to be performed by a specific agent within a given time window during which the required resources for this task are allocated to this agent.

In addition, we assume that the initial resource allocation is such that some allocated resources are not fully used by agents, i.e. an agent could perform additional tasks during given time windows of it's initial task schedule. So, the main idea of this work is to consider that agents pool their *residual resources* in order to execute tasks for other agents and to enable task reallocation. The main goal of each agent is to increase the value of their *residual resources* by means of monetary valuation and also to be able to use cheap resources to delegate the achievement of one or more punctual tasks.

Our approach aims to cover a wide range of possible applications such as computer network, grid computing, shipping logistics, transportation network, supply chain, manufacturing system, search and rescue mission, etc...

For example, if we consider a computer network conducted by a multi-agent system and with a defined resource allocation, during the execution of the task schedule some failure may occur. As a consequence some tasks may become unfeasible. Moreover some resources may be allocated to agents and may not be fully used by them, i.e. *residual resources*. Our idea is that by pooling those *residual resources* it could be possible to achieve some of the unfeasible tasks with a new ad-hoc task allocation.

As we focus on the case of tasks reallocation during the execution of a defined schedule, the main constraint of our problem is that we cannot modify the initial resource allocation. We must rather use this allocation in order to assign the execution of additional tasks to agents based upon their allocated resources and initial task schedule (i.e. without modifying the agents initial task schedule).

[1] Univ Lille Nord de France, F-59000 Lille, Franc; UVHC, LAMIH, F-59313 Valenciennes, France; CNRS, FRE 3304, F-59313 Valenciennes, France; email: {gael.hette}{sylvia.estivie}{emmanuel.adam}{rene.mandiau}@univ-valenciennes.fr

2.2 State of the art

A possible approach to solve resource allocation and task scheduling problems in distributed systems is to consider and formalize such problems as Distributed Constraint Satisfaction Problem (*DCSP*) in which variables and constraints are distributed among multiple agents. As described by M. Yokoo [16], resolution of *DCSP* are mainly achieved by use of distributed algorithms. Like algorithms for solving *CSP*, *DCSP* algorithms can be divided into two groups : search algorithms (i.e. backtracking and iterative improvement) and consistency algorithms (arc and path consistency). *DCSP* algorithms are efficient to solve large-scale resource allocation problems within a reasonable amount of time [16]. But as noted by Harvey et al. [8] *DCSP* algorithms use ordering over the variable which induces an order among agent authority and implies priority over resource allocation between agents. Thus we ignore this approach, since for the purpose of our work, we do not want any authority order between agents.

The remainder of this section starts with a presentation of one major problem where the need of task reallocation could arise in case of disturbance or failure during the execution of a defined schedule. Thus, under some assumptions, it could be possible and profitable to apply the pooling of *residual resources* in order to resolve such kind of problem. Furthermore, to the best of our knowledge there is no study in the multi-agent literature which addresses the problem of task reallocation during the execution of a schedule and without any modification of the initial resource allocation. Therefore we give an overview of different approaches for resource allocation and tasks scheduling which deal with problems that are closely related to our (Section 2.1). We present these approaches as possible leads and insights to cope with task reallocation without modifying the initial resource allocation and task schedule of agents.

The *Vehicle Routing Problem with Time Windows* (*VRPTW*) [10] is a major problem of interest as it can be used to model many applied problems. This problem has been studied widely in both operational research [14, 3] and multi-agent systems literature [5, 7]. The *VRPTW* concerns the design of least cost routes over a fleet of vehicles, in order to deliver goods from a central depot to a set of geographically scattered points during given time windows[2]. The *VRPTW* is considered as *static* in the case that all relevant informations are known beforehand. Otherwise the problem is considered as *dynamic* and is referred as *Real-Time VRPTW* [11, 6].

In the multi-agent literature, the market based approach is the main approach adopted to solve VRPTW. A common model is based on a manager agent to represent the shipping company and a set of agents to represent trucks of the company. The task allocation process comprises a sequence of successive or parallel allocation procedures where delivery orders are proposed to the truck agents by the manager agent. These allocation procedures are often based on the Contract Net Protocol [4, 5] or on the Vickrey auction model [15, 7, 9].

Fischer et al. [5] presented a multi-agent system simulation framework where agents interactions are driven by negotiation protocol leading to different task scheduling mechanisms. They produced an implementation and experimental results of their approach to solve the *Vehicle Routing Problem with Time Windows* (*VRPTW*). This study focuses on a problem that is similar to ours in the sense of resource pooling, as it proposes a peer-to-peer negotiation between companies to buy and sell free loading capacities (i.e. *residual resources*). But these considerations differ with ours because this negotiation does not occur in the case of task reallocation. As a matter of fact, a task reallocation procedure is used to overcome disturbance during the execution of a plan. This procedure allocates the unfeasible tasks of an agent to other agents of the same company. Considering the constraints of our problem, the outcomes of this procedure are interesting. Indeed, in the best case the procedure can lead to local rescheduling of agents plans, whereas in the worst case it leads to a more global rescheduling.

Another multi-agent system simulation framework for resource allocation and task scheduling which is based on auction has been proposed by V. Godoretski et al. [7]. They produced an implementation and experimental results to solve the *VRPTW*. Their study differs with our problem because it focuses on the case of one shipping company and there is no consideration about dealing with disturbance during the execution of a plan.

Hoen et al. [9] studied the concept of decommitment in a large scale logistics setting of competing transportation companies. The action of decommitment consists in replacing a won contract by another (more profitable) contract. As in our problem, their approach deals with a set of competitive companies and their decommitment strategy can be seen as a process of tasks reallocation with few modifications of agents task schedule. The decommitted tasks are reallocated among the agents of the company which holds the contract, whereas by pooling *residual resource* we aim to reallocate tasks among agents of all companies. In this study the objective of decommitment is to increase the profit, but we consider that a similar decommitment strategy could be used to overcome disturbances during the execution of multiple *VRPTW* solution instances.

Sadri et al. [13] proposed an approach to solve the *Temporal Resource Reallocation Problem* (*TRRP*), based on multiple stages of negotiation using dialogue moves in a society of self-interested agents sharing resources. The context of *TRRP* is such that agents aim to accomplish tasks that require a specific resource, have a specific duration and must be done within a given time window. At start, the initial distribution of resources is such that agents do not necessary own the resources required by their activities. Thus agents negotiate to exchange the possession of resources over specific time windows. The negotiation process leads to compute a resource allocation and schedules such that agents can perform their tasks within the given time constraints. P. Alexopoulos et al. [1] extended the work of Sadri et al. [13], by modifying the protocol and policy of the third negotiation stage in order to allow multiple resources exchange dialogues. In regard with our problem these approaches are interesting because they aim to compute a resource allocation within time windows. The difference between these two works and our problem is that under the *TRRP* context agents exchange resources and modify their schedule. As a matter of fact, it could be interesting for us to adapt these negotiation protocols such that agents exchange and/or delegate tasks execution within specified time windows and according to their *residual resources*.

3 Computational model

We consider a multi-agent system populated by n agents $A_1, ..., A_n$. In order to enable agents to perform an argumentation based negotiation protocol (Section 3.1), we use a Belief-Desire-Intention model [12] inspired by Sadri et al. [13] to define the knowledges of agents. Thus, the knowledges of an agent A_i are represented by a tuple $< \mathcal{B}_i, \mathcal{I}_i, \mathcal{R}_i, \mathcal{O}_i, \mathcal{D}_i, \mathcal{DS}_i, \mathcal{G}_i >$ with :

[2] Delivery locations must be visited only once and by exactly one vehicle during a given time window. Each route starts and ends at depot/warehouse. The total load of a vehicule for a given route does not exceed its capacity.

- $\mathcal{B}_i$: *beliefs* (negotiation policy, informations about self and other agents),
- $\mathcal{I}_i$: *intentions* (plan, task schedule of agent i),
- $\mathcal{R}_i$: allocated *resources* ,
- $\mathcal{O}_i$: $\{O_{i1}, ..., O_{iu}\}$ a finite set of *task offers* of agent i,
- $\mathcal{D}_i$: $\{D_{i1}, ..., D_{iv}\}$ a finite set of *task demands* of agent i,
- $\mathcal{DS}_i$: *store* of past dialogues,
- $\mathcal{G}_i$: agent *goals*.

In our model, the desires that an agent wishes to satisfy corresponds to its task demands. The intentions $\mathcal{I}_i$ of an agent A_i represent its task schedule. This schedule is an ordered set of tasks. Each task is specified by a time window and its required resources. As Sadri et al. [13] we assume that each task needs only one resource and that all tasks requiring the same resource have disjoint time windows. In regard with our problem, the initial resource allocation is such that each agent of the system owns all the required resources of its initial task schedule. In addition, the task schedules of agents are defined such that, during some time periods, all allocated resources are not needed by the agents (i.e. *residual resources*). We assume that during such time periods agents can perform additional tasks requiring these *residual resources*. Thus, agents dispose of a set of *task offers* as a mean to propose to other agents to execute tasks for them. According to the dynamic of the system, some disturbances occur during the execution of agents task schedules. In our model we define a disturbance as a loss of one or more initially allocated resources of an agent. Therefore, agents incurring disturbances cannot perform the tasks requiring such unavailable resources. To overcome such situation agents dispose of a set of *task demands* which define the set of tasks that an agent cannot perform and could delegate to other agents. Agents create a *task demand* for each scheduled tasks it cannot perform due to the loss of a resource. At start this set of *task demands* is empty. The content of this set evolves along the dynamic of the system.

Given an agent A_i, we define each elements of $\{O_{i1}, ..., O_{iu}\}$ and $\{D_{i1}, ..., D_{iv}\}$ as finite sets of data respectively describing the *residual resource* associated to a *task offer* and the resource associated to a *task demand* of A_i. Each elements of $\{O_{i1}, ..., O_{iu}\}$ and $\{D_{i1}, ..., D_{iv}\}$ are composed of the following data set $(A_{rc}, R_{rc}, T_{rc}, V_{rc}, P_{rc}, Loc_{rc}, Pref_{rc})$, where :

- A_{rc} : the agent offering or needing resource,
- R_{rc} : the type of the offered or needed resource,
- T_{rc} : a time interval defining a time constraint over the period when the resource is available or needed to perform a task delegation,
- V_{rc} : the volume or amount of the offered or needed resource,
- P_{rc} : the price (a monetary valuation) of the offered or needed resource,
- Loc_{rc} : the location of the offered or needed resource,
- $Pref_{rc}$: a finite set of preferences regarding the use of the resource (Section 3.2).

Depending on the type of problem addressed, some of the above constraints are not necessarily affected, e.g. for a computer network problem the Loc_{rc} constraint may be useless. Our brief state of the art along task and resource reallocation problems lead us to think that temporal constraints are a common and crucial point of interest in such problems. So in our approach we consider that temporal constraint is the only necessary constraint for the core class of task allocation problems addressed by our framework. Thus we assume that the following constraints are optional and used in particular cases :

- V_{rc} : used in *VRPTW*[3] as a constraint on freight load, or in *TRRP*[4] as the constraint on required resource type;
- P_{rc} : to address problems with a market based approach or if the addressed problem involves selfish agents;
- Loc_{rc} : used if the addressed problem concerns situated agents (ie. *VRPTW*).

In the reallocation procedure (Section 3.1), agents try to find the best *task offer* corresponding to their needs by asking other agents about their *task offer*. The reallocation procedure also involves a negotiation step. In order to perform the reallocation procedure, agents must communicate according to the application context of the addressed problem. Hence, we distinguish two possible alternatives over agent's communication ability :

- the communication between agents are constrained, such that at a given time an agent is only able to communicate with a restricted number of other agents (e.g. communication can be constrained by topological consideration);
- there is no communication constraint, agents can communicate any other agent at any time.

In the context of our study, we assume that we are in the second case and there is no communication constraint between agents.

3.1 Task reallocation process

In the context of our problem, we consider a multi-agent system which evolves over time. Agents have to perform tasks in a dynamic environment, and we assume the following *system cycle* :

1. *Environment dynamics* : the environment evolves according to the defined rules of the system dynamics. *Disturbances* occur at random, i.e. some resources become unavailable to one or several agents.
2. *Perception step* : agents get perceptions from the environment. These perceptions concern the current availability of the initially allocated resources. Agents facing disturbances update their knowledges in regard with their unavailable resources. As a matter of fact, these agents generate and add new *task demands* to their own set of *task demands*.
3. *Reasoning step* : agents facing disturbances (i.e. agents with a non empty set of *task demands*) create a list of agents which may propose each one or more appropriate *task offers*.
4. *Communication step* : agents facing disturbances ask the previously selected agents about their *task offers*. If one or more *task offers* fulfill the needs of a *task demand*, the agent choice between the different offers is based upon its preferences model (Section 3.2). If there is no appropriate *task offer*, the agent performs a negotiation process with agents offering less appropriate *task offers*.
5. *Action step* : agents performs their scheduled tasks in regard with the availability of the required resources.

We propose to decompose the task reallocation process of an agent A_i into three main steps. The first step of the reallocation process occurs during the reasoning step of agent A_i. During this step A_i computes a list of agents to query about their *task offers*. This list of agents is determined upon the beliefs of A_i about other agents. These beliefs about other agents have been acquired during former interactions with agents.

[3] Vehicule Routing Problem with Time Windows
[4] Temporal Resource Reallocation Problem

The second step of the reallocation process occurs during the communication step. During this step, A_i queries all the previously selected agents about their *task offers*. In response, each of the selected agents compute their current set of *task offers* without any consideration of the *task demand* of agent A_i. Then, all the selected agents reply to A_i by sending their current set of *task offers*. Thus A_i retrieves from these agents a set of acceptable *task offers* in regard with its needs (i.e. the preferences of A_i *task demand*). If the set of acceptable offers is empty, then agent A_i can perform the following actions:

- restart the reallocation process and determine a new list of agents according to its new beliefs about others in order to ask new agents about their *task offers*,
- negotiate upon time constraints, with agents offering less acceptable *task offers*.

The third step of the reallocation process occurs if the set of acceptable *task offers* is not empty. During this step the agent A_i chooses, with respect to the preferences of its *task demand*, the best offer from the set of acceptable *task offers*. Then A_i concludes the delegation of its task with the agent proposing the chosen *task offer*.

3.2 Agents preferences representation

In order to define how agents should make a decision over a set of acceptable solutions, we introduce a set of ad hoc preferences. Preferences are a mean to express the satisfaction of an agent towards a specific offer. During the second step of the reallocation process it is up to the agent A_i, who is missing a resource, to perform a research heuristic as a way to find acceptable offers. To reduce the computational cost of this heuristic, the selection of acceptable offers is only based upon agent A_i satisfaction, and as a matter of fact only upon A_i preferences. As a result we define a class of preferences specific to the *task demand* of an agent as follows :

- T_{pref} : preferences among time constraints of a set of offers. This preference enables agents to express how much a task delegated to another agent can be postponed.
- P_{pref} : preferences among prices of different *task offers*. For instance, it determines if the agent chooses the cheapest priced or the average priced offer.
- PT_{pref} : preference between price and time period proposals, over a set of acceptable solutions. This enable to define if an agent chooses to focus on price constraint rather than time constraint or vice versa.
- Loc_{pref} : preferences over the location of different *task offers* to choose with.
- Co_{pref} : preferences over a specific group of agents to work with. Thus, between different acceptable *task offers* an agent chooses the offer made by an agent specified by Co_{pref}.

In regard with all the above different preferences associated to a *task demand*, a major question of interest is how to deal with this set of preferences to determine the choice of an agent between two or more acceptable *task offers*. In a future work, we will define a procedure to aggregate this heterogeneous preferences.

During the second stage of the reallocation process, agents exchange dialogues of negotiation by pairs. Both agents participating in the dialogue use a negotiation policy to formulate dialogue moves. The negotiation policy of an agent is based upon its preferences regarding its *task demand* or its *task offer*. As needed by an agent involved in a negotiation dialogue, we define the following specific class of preferences :

- T_{pref} : preferences over the time period to execute the offered task. Agent negotiate over the time period during which the task should be executed,
- Loc_{pref} : agent negotiate over the location of the offered *residual resources*,

4 An application for logistic networks

We consider a framework which aim to manage pooling capacity for transportation and storage of goods/freight between different logistic companies sharing their resources. Unexpected events may occur during the execution of the contracts (transportation or storage), e.g. truck breakdown or warehouse failure. Such unexpected events must be solved by finding new transportation or storage means within the pooled resources of the framework, and by creating new transportation plans.

We consider a multi-agent system which represent a pooling capacity framework. Each agent represent transportation or storage mean, e.g. trucks or warehouses. The environment is described as a spacial grid divided into small non overlapping areas. Agents are localized by their position in a unique area of the spatial grid. Time is considered continuous.

There is no communication constraint between agents. Each agent can handle a specific bounded capacity (volume and weight) of freight. During the execution of the system agents' freight load evolve. Actually agents capacity are not necessary fully loaded, so unused transportation and storage capacity are still available.

To cope with unexpected events agents must be able to find new means of transportation and/or storage within the agent society. Thus they must communicate with each others to find agents with corresponding residual capacity and elaborate new plans.

5 Conclusion

In this paper, after a presentation of the problem addressed by our study and a short state of the art, we proposed a computational model of the multi-agent system that allows agents to negotiate the reallocation of tasks. We introduced the possible exploitation of residual resources in a multi-agent system where agents dispose of non shareable resources to execute a defined task schedule. To overcome disturbances during the execution of such schedule, we propose to reallocate tasks to agents over time periods where they can perform additional tasks without modifying their initial schedule. During the reallocation process agents try to delegate to other agents the execution of tasks they can no longer perform due to a loss of resources.

There are many lines of future research related to this work. A major question concerns the definition of an aggregation procedure for the heterogeneous set of agents' preferences. Another issue that is not covered in this work and that we will study in the future is the agents negotiation protocol and policy based on the specific class of preferences proposed in Section 3.2. We will also intend to refine the second step of the reallocation procedure. At this step of the procedure, if an agent is unable to find any acceptable *task offers*, it can restart the search of acceptable *task offers* or negotiate with agents offering less acceptable *task offers*. The issue of this refinement is to enable agents to determine if they either initiate a negotiation process or restart the search of *task offers*. It is obvious that in certain cases it is not efficient to perform a negotiation process. Thus we will have to exhibit such cases and find a mean to avoid agent to negotiate in these cases. Another issue of future work is to implement our computational model and propose an experimental evaluation of the reallocation procedure and its' computational complexity based on the number and the size of the messages exchanged between agents.

ACKNOWLEDGEMENTS

We would like to thank the referees for their comments, which helped improve this paper. The present research work has been supported by International Campus on Safety and Intermodality in Transportation, the Nord-Pas-de-Calais Region, the European Community, the Regional Delegation for Research and Technology, the French Ministry of Higher Education and Research, the French National Center for Scientific Research, the Ile-de-France Region, the "Ministre de l'conomie, de l'Industrie et de l'Emploi", and ADVANCITY. The authors gratefully acknowledge the support of these institutions.

REFERENCES

[1] Panos Alexopoulos, 'Resource sharing in multi-agent systems through multi-stage negotiation', in *AIAI*, eds., Christos Boukis, Aristodemos Pnevmatikakis, and Lazaros Polymenakos, volume 247 of *IFIP*, pp. 57–64. Springer, (2007).

[2] Yann Chevaleyre, Paul E. Dunne, Ulle Endriss, Jérôme Lang, Michel Lemaître, Nicolas Maudet, Julian A. Padget, Steve Phelps, Juan A. Rodríguez-Aguilar, and Paulo Sousa, 'Issues in multiagent resource allocation', *Informatica*, **30**(1), 3–31, (2006).

[3] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and F. Soumis, 'VRP with time windows', 157–193, (2001).

[4] R. Davis and R. G. Smith, 'Negotiation as a metaphor for distributed problem solving', *Artificial Intelligence*, **109**, 20–63, (1983).

[5] Klaus Fischer, Brahim Chaib-draa, Jörg P. Müller, Marcus Pischel, and Christian Gerber, 'A simulation approach based on negotiation and cooperation between agents: A case study', *IEEE Trans. on Systems, Man, and Cybernetics*, **29**, 531–545, (1999).

[6] Zheng gang Dan, Lin ning Cai, and Li Zheng, 'Real-time distribution system based on multi-agent', in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Volume I, IMECS '08, March 19 - 21, 2008, Hong Kong*, eds., S. I. Ao, Oscar Castillo, Craig Douglas, David Dagan Feng, and Jeong-A Lee, Lecture Notes in Engineering and Computer Science, pp. 218–221. International Association of Engineers, Newswood Limited, (2008).

[7] Vladimir I. Gorodetski, Oleg Karsaev, and Victor Konushy, 'Multi-agent system for resource allocation and scheduling', in *CEEMAS*, eds., Vladimír Marík, Jörg P. Müller, and Michal Pechoucek, volume 2691 of *Lecture Notes in Computer Science*, pp. 236–246. Springer, (2003).

[8] Peter Harvey, Chee Fon Chang, and Aditya Ghose, 'Support-based distributed search: a new approach for multiagent constraint processing', in *AAMAS*, eds., Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone, pp. 377–383. ACM, (2006).

[9] Pieter Jan't Hoen and J. A. La Poutré, 'A decommitment strategy in a competitive multi-agent transportation setting', in *AMEC*, eds., Peyman Faratin, David C. Parkes, Juan A. Rodríguez-Aguilar, and William E. Walsh, volume 3048 of *Lecture Notes in Computer Science*, pp. 56–72. Springer, (2003).

[10] Gilbert Laporte and Ibrahim H. Osman, 'Routing problems: A bibliography', *Annals of Operations Research*, **61**(1), 227–262, (1995).

[11] H. N. Parfaits, 'Dynamic vehicle routing : status and prospects', *Annals of Operations Research*, **61**(1), 143–164, (1995).

[12] Anand S. Rao and Michael P. Georgeff, 'BDI agents : From theory to practice', in *ICMAS*, eds., Victor R. Lesser and Les Gasser, pp. 312–319. The MIT Press, (1995).

[13] Fariba Sadri, Francesca Toni, and Paulo Torroni, 'A multi-stage negotiation architecture for sharing resources amongst logic-based agents', in *UKMAS, 2002*, (2002).

[14] Marius M. Solomon, 'Algorithms for the vehicle routing and scheduling problems with time window constraints', *Operations Research*, **35**, 254–265, (1987).

[15] W. Vickrey, 'Counterspeculation, auctions and competitive sealed bids', *Journal of Finance*, **16**, 8–37, (1961).

[16] Makoto Yokoo, *Distributed constraint satisfaction: foundations of cooperation in multi-agent systems*, Springer-Verlag, London, UK, 2001.