# A Task-Swap Negotiation Protocol Based on the Contract Net Paradigm

Matteo Golfarelli, Dario Maio, Stefano Rizzi

DEIS, CSITE - Università di Bologna
Viale Risorgimento, 2
40136 Bologna, Italy

## Abstract

In this paper we propose a market-like negotiation protocol which extends the well-known Contract Net Protocol to cope with domains where task swap is the only possible type of contract. The agents we consider are heterogeneous and self-interested. The absence of an explicit support for utility transfer determines interesting implications on efficiency and stability; in particular, we propose different design alternatives induced by different compositions of the announcement and the bid, and discusses the strategies supported. Task swapping is more feasible than task selling for many realistic applications; symbolic path planning in autonomous robotic agents is the application on which we focus the description of our approach.

## 1 Introduction

When several agents are placed within an environment to carry out tasks, they should coordinate their activities in order to detect and resolve the conflicts which may arise in planning [Kakehi and Tokoro, 1993; Lander and Lesser, 1993]; besides, they may cooperate to decrease the task execution costs. The cooperation techniques to be adopted depend strongly on the nature of the agents: a *benevolent agent* tends to maximize the global utility of the agent society, whereas a *self-interested agent* is inclined towards maximizing its own profit and thus is open to cooperation only if it is an advantage for itself. Different approaches to this problem can be found in the DAI literature [e.g., Rosenschein and Zlotkin, 1994]; the research is primarily directed towards determining the negotiation algorithms which the agents can adopt to reach a common agreement in a distributed fashion.

The Contract Net Protocol (CNP) [Smith, 1980] is an approach to negotiation in multi-agent systems, inspired by a market-like model of negotiation [Malone *et al.*, 1988]. Each agent can formulate a *bid* for each *announcement* received; the contract will be *awarded* to the agent which sends the best bid. The CNP was initially used to allocate tasks over a distributed network of sensors, and has been extended and applied within several contexts with reference to benevolent agents [e.g., Malone *et al.*, 1988; Sen, 1994].

Self-interested agents have been considered by Lesser and Sandholm, who applied the CNP to the electronic market [Sandholm, 1993], formalized the basic protocol and introduced *bounded rationality* and *levels of commitment* [Sandholm and Lesser, 1995]. In that framework, tasks are always sold for money.

In this paper we propose a negotiation protocol which conforms the contract net paradigm to environments where the only possible type of contract is the swapping. The agents we consider are self-interested and have limited communication capabilities; they may have different skills and cost evaluation functions, and must carry out different missions. Thus, the utility of a given swap may differ substantially for the different agents. Since no agent owns models of the others, and there is no money to establish a common cost metric, an agent has no means of estimating the utilities for the other agents. This causes the swap-based protocol to be more stable than the sale-based one.

Many realistic applications may benefit from adopting swap-based contracts instead than sale-based ones. In particular, in this paper we discuss how autonomous robotic agents can decrease the costs for executing their missions by swapping the tasks they have been assigned.

The paper is organized as follows. In Section 2 the background is presented; in particular, the approach to environmental knowledge representation, the assumptions on the agents, the task formalization, and the path-planning algorithm adopted are briefly discussed. In Section 3 the negotiation protocol is described and different design alternatives are examined; in Section 4 the different strategies compatible with the protocol are discussed. In Section 5, some experimental results are presented and the conclusions are drawn.

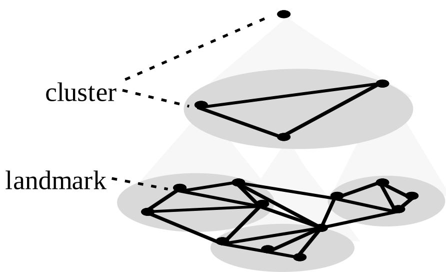


Figure 1. Knowledge representation.

## 2 The Scenario

Our research on DAI is placed within the robotics field. Thus, by the term *agent* we denote a robot capable of moving autonomously within a (partially) known environment. Each agent is capable of performing specific actions (for instance, carrying objects) and owns a private, possibly incomplete description of the environment. Knowledge of the environment is organized on multiple *layers* at different abstraction levels [Maio and Rizzi, 1996]; each is structured as a graph whose vertices and arcs represent, respectively, places and connections between them. Within the graph at the lowest abstraction level, vertices correspond to *landmarks* (distinctive or serviceable sites in the environment); within the graphs at the higher levels, each vertex corresponds to a *cluster*, i.e., a connected subgraph of the graph at the level below (see Figure 1) [Maio *et al.*, 1996].

Path planning in robotic agents is the search for a route in the environment which enables an agent to achieve a set of inter-related goals. In symbolic approaches, path planning is carried out on a symbolic description of the environment; the main interest is not in achieving motion in the physical space, but in determining, on a network of places and routes, a sequence of places to be visited in order to carry out a set of tasks [Causse and Crowley, 1994; Dean *et al.*, 1988; Timpf *et al.*, 1992]. Two kinds of constraints are usually considered: constraints on the resources to be used and temporal constraints related to the execution of tasks and to resource availability. Both topics are widely discussed in the literature; see for instance [Allen, 1991; Ghallab and Vidal, 1996; Wilkins, 1988; Ghallab and Laborie, 1995]. In our approach, each agent is assigned a mission consisting of a set of tasks, which it must carry out successfully. We assume that each task:

- entails an *action* (for instance, commute a switch or empty an ash-tray).
- must be executed on a given *resource*. By the term resource we mean a landmark/cluster where the action must be executed (for instance, "*the director's computer*"); a resource may also be defined as a set of landmarks/clusters: in this case, the action may be executed indifferently on either of them (*resource type*; for instance, "*any computer*").
- may be constrained to be executed within a given *time window* (for instance, because the resource required is not available outside that time window).
- may be related by *precedence constraints* to the other tasks (before a task can be executed, another task must have been carried out).
- may require and/or produce one or more *objects* (for instance, documents and floppy disks).

This is an example of a set of tasks, expressed in natural language, for an office-agent: "*Pick up a document from the secretary's desk, stamp it in the director's office and take it back to the secretary; take a copy of the stamped document to the archive. The director leaves at 11 a.m. and the archive is open from 10 a.m.*". Here, the document and its copy are objects produced and required; "*the secretary's desk*", "*the director's office*", "*any photocopier*" and "*the archive*" are resources; "*pick up*", "*take*", "*stamp*" and "*photocopy*" are actions; "*until 11 a.m.*" and "*after 10 a.m.*" denote time windows; the fact that the document must have been stamped before it is copied implies a precedence constraint.

In order to carry out the tasks it has been assigned, each agent must plan a path which visits the resources involved by respecting the constraints imposed. The agent evaluates a path by means of a *cost* function which may express, for instance, its length or the time/fuel required to follow it. We will denote with $c_A(p)$ the cost of path $p$ as measured by agent $A$. The primary goal of each agent is to execute its tasks at the lowest possible cost. [Maio and Rizzi, 1996] proposes a planning algorithm capable of generating a low-cost path, *path* ($S$), to execute a given a set of tasks $S$; this algorithm is used by each agent, in the negotiation protocol explained in Section 3, to plan its path and to evaluate the utilities of the task swaps.

Several robotic agents may move within the same environment. We assume that they are heterogeneous and self-interested, do not share any physical memory and are not coordinated by any central supervisor. Agents communicate with each other by broadcasting messages; the communication range is limited to a radius $\rho$. We assume that agents own neither money nor any other explicit means for utility transfer. Moreover, different agents may adopt different cost measures: in fact, the protocol we propose never compares two costs calculated by different agents.

## 3 The Swap-Based Negotiation Protocol

The scenario outlined in Section 2 imposes some constraints on task negotiation:

- Since the agents do not own money, they cannot sell tasks; thus, the only way an agent can decrease its execution cost is to swap tasks with other agents.
- The self-interested nature of the agents and the reservedness of the tasks to be carried out leads to minimizing the amount of information shared.
- Since the communication radius is finite, agents will not receive, in general, all the messages broadcasted by others. This has two main consequences. Firstly, an agent can hardly monitor the complete history of the past negotiations between the others, hence, it cannot make reliable assumptions on their goals. Secondly, an agent that has bound itself to a contract does not know whether it will be enabled to revoke that contract (*decommitment*) or not. Thus, we assume that agents do not take past and future negotiations into account.
- Negotiations must take place during task execution, so they should not have high computational complexity. The agents are assumed to be *bounded rational*, i.e., their computational resources are assumed to be expensive and/or limited [Sandholm and Lesser, 1995].

In this section, after introducing the metrics we adopt to evaluate task costs and swap utilities, we outline the negotiation protocol we have designed to satisfy the assumptions above and propose different alternatives for formulating announcements and bids. Finally, referring to the path-planning application, we discuss some issues concerning the swap-oriented processing of tasks which include precedence and object constraints.

### 3.1 Marginal Cost and Utility

Let $A$ be an agent and $S$ be the set of its current tasks. Given a task $s$, we define its *relative marginal cost* (RMC) for $A$ as the cost paid by $A$ to execute $s$:

$$rmc\ (s, A, S) = \begin{cases} c_A(path\ (S)) - c_A\big(path\ (S - \{s\})\big), & s \in S \\ c_A\big(path\ (S \cup \{s\})\big) - c_A(path\ (S)), & s \notin S \end{cases}$$

The RMC of a task for an agent that is not capable of performing the action involved is considered to be infinite.

In sale-based protocols, the RMC is used to give a monetary evaluation of the tasks. In a swap-based protocol, the RMC may be useful to select the tasks(s) to be included in the announcements (see Section 4); on the other hand, the convenience of a swap is evaluated by measuring its utility.

The *utility* for agent $A$ in swapping its task $s_2 \in S$ with task $s_1$ assigned to agent $B$ is measured as:

$$uti\ (s_1, s_2, A, S) = c_A(path\ (S)) - c_A\big(path\ (S \cup \{s_1\} - \{s_2\})\big)$$

It is straightforward to verify that

$$uti\ (s_1, s_2, A, S) = \big(rmc\ (s_2, A, S) - rmc\ (s_1, A, S - \{s_2\})\big)$$

A swap is said to be *individual rational* for an agent if its utility is positive.

It should be noted that both the RMC of a given task and the utility of a given swap may differ substantially for the different agents according to their capabilities, their current tasks and the instructions received which may lead them to privilege, for instance, time rather than energy consumption. Thus, an agent has no means of estimating either the RMCs or the utilities for the other agents.

Using the swap affects the efficiency of negotiations, by introducing some possibilities which would not be individual rational if the sale were used and which allow local maxima in the utility functions to be overcome. Consider for instance the following table:

| *Tasks* | *RMC for agent A* | *RMC for agent B* |
| --- | --- | --- |
| $s_1$ | 4 | 3 |
| $s_2$ | 3 | 4 |
| $s_1$ & $s_2$ | 10 | 10 |

Agents $A$ and $B$ must execute tasks $s_1$ and $s_2$, respectively, with global cost 4+4=8. No sale contract can be agreed upon, since for each agent the cost for executing both tasks is 10>4. The only possible solution is the task-swapping, which leads to a global cost equal to 3+3=6.

Of course, there are cases in which the only individual rational negotiation is the sale. Thus, adopting the swap as the only type of negotiation is justified only if, within the domain considered, no explicit mechanism for utility transfer exists.

### 3.2 The Negotiation Cycle

Basically, each agent operates a *negotiate, plan and execute* cycle. During each negotiation session, each agent tries to exchange all its current tasks once. When the session terminates, the agent plans a path to execute its current tasks and starts to follow it. After the first resource has been reached and the corresponding task has been executed, a new negotiation session is launched on the remaining tasks.

In the sale-based protocol, different levels of commitment are defined by associating each task with a monetary penalty which an agent must pay if it decides to recede from a subscribed contract; thus, an agent can participate in several negotiations simultaneously. In a swap-based protocol, such a penalty cannot be applied; besides, each negotiation is influenced by the results of the previous ones, since the utility of a swap depends on the tasks already added or removed. For these reasons, and since an agent is bound to a swap at the moment it broadcasts a bid, we assume that no agent can bid more than one announcement at a time.

Like the CNP, our swap-based negotiation protocol is based on three phases:

**Announcement**. The announcing agent formulates an announcement and broadcasts it to the other agents. An announcement consists of one or more tasks which the announcer is interested in exchanging.

**Bid**. Each agent receiving the announcement formulates a bid and broadcasts it to the announcer. The bid associated to an announcement consists of one or more tasks, owned by the bidder, which the bidder is interested in exchanging with the tasks included in the announcement. A swap is proposed only if it is individual rational. The bidder cannot bid other announcements nor can it start a new announcement until it receives either an award or an acknowledgment from the announcer.

**Award**. The announcer collects the bids for a fixed time. When this time expires the announcer determines, among all the bids received, the swap having the highest utility. If the utility is positive, then the swap is individual rational for both the announcer and the bidder; an award is broadcasted to the winner and the swap is confirmed. An acknowledgment is broadcasted to all the non-winner bidders. The negotiation session is over.

### 3.3 Formulation of Announcements and Bids

Based on the general framework we have outlined, different design alternatives are possible; their properties are summarized in Table I.

| | *Design* | *Stability* | *Complexity* | *Swap Probability* | *Utility* |
|---|---|---|---|---|---|
| #1 | One announcement / One bid | ↑ | $n_1+n_2+1$ | ↓ | ↓ (in favour of the bidder) |
| #2 | One announcement / Many bids | | $n_1+n_2+m_2$ | | (in favour of the announcer) |
| #3 | Many announcements / One bid for each announc. | | $n_1+m_1 \times n_2+m_1$ | | (in favour of both) |
| #4 | Many announcements / Many bids for each announc. | | $n_1+m_1 \times n_2+m_1 \times m_2$ | | (in favour of the announcer) |

Table I. Design alternatives for the swap-based CNP, evaluated in terms of their stability, complexity, utility and of the probability of the swap. Computational complexity is expressed in function of the number of path-planning problems to be solved: $n_1$ and $n_2$ are the number of current tasks of the announcer and of the bidder, respectively; $m_1$ is the number of tasks announced and $m_2$ the average number of tasks bid for each task announced.

Let $n_1$ and $n_2$ be, respectively, the number of current tasks of the announcer and of the bidder; let $m_1$ be the average number of tasks included in an announcement and $m_2$ the average number of tasks bid for each announced task. The announcer calculates $n_1$ RMCs to find out the most advantageous task(s) to be swapped; the bidder calculates $m_1 \times n_2$ utilities to determine the best swap(s); the announcer calculates $m_1 \times m_2$ utilities to determine if the swap(s) proposed by the bidder is(are) individual rational. Thus, in general, the computational cost in terms of the number of path-planning problems to be solved is

$$n_1 + m_1 \times n_2 + m_1 \times m_2$$

In cases #1 and #2 it is $m_1$=1; in cases #1 and #3 it is $m_2$=1.

In case #1, both the announcement and the bid consist of one task. This protocol presents a dominant strategy, since the only choice each of the two agents can make is to propose its most expensive task. The computational cost in terms of the number of path-planning problems to be solved is low. The probability that the announcer and the bidder come to an agreement on a swap is low, since only a few swaps are evaluated; besides, since neither of the two agents is capable of computing the utility function of the other, it cannot predict whether the swap will be individual rational for the other. The low number of combinations considered also reduces the average utility of the swaps; however, the utility tends to be higher for the bidder, the only one that can choose the task to propose by computing and maximizing the swapping utility.

The other alternatives can be evaluated qualitatively in a similar manner. In particular, alternatives #2, #3 and #4 do not offer any dominant strategy since the bidder can formulate its bid in different ways. For instance, it can decide to propose only the swaps offering higher utilities: in this case, the average utility of the swaps for the bidder is increased, but the probability of a successful negotiation is decreased. In alternatives #3 and #4, also the announcer can choose between different strategies: announce all its current tasks, or announce only the most expensive one(s).

However, we consider alternative #2 (*One announcement /Many bids*) to be the one which best fits the assumptions made in Section 2. In fact, it offers the best trade-off between complexity and utility, where complexity deserves a particular attention due to the bounded rational nature of the agents. The number of path-planning problems to be solved is linear in the number of tasks to be executed; adopting alternatives with higher complexity would lead to increasing the time required to formulate the bid, hence, also the stand-by time of the agents participating in the negotiation. We have verified experimentally that the extra utility produced by "long" negotiation does not balance the loss due to the reduced number of negotiations made.

In Section 4 we discuss the properties of the chosen design alternative and the strategies it supports.

### 3.4 Exchanging Tasks with Precedence Constraints

Exchanging a task having one or more precedence constraints requires each constraint to be solved. Consider an agent *A* that has been assigned a couple of tasks $s_1$ and $s_2$, and let $s_1$ be required to be carried out before $s_2$. If $s_2$ is negotiated and assigned to another agent *B*, an undesired dependence link is created between the two paths planned by *A* and *B*. The simplest solution to this problem is to transform the precedence constraint into a temporal constraint which, due to its absolute nature, makes the two paths independent of each other. Before negotiating, *A* determines an execution plan for its tasks by applying the *path* () algorithm; this plan exactly defines at what time each task will be carried out. Let *t* be the time scheduled for $s_1$; when proposing $s_2$, *A* narrows its time window to $]t,\infty[$, thus forcing *B* to execute $s_2$ after $s_1$. Similar considerations can be made if $s_1$ is negotiated.

### 3.5 Exchanging Tasks with Object Constraints

In Section 2 we have assumed that a task may require and/or produce one or more objects. From the point of view of path planning, object constraints on tasks entail precedence constraints. Translating an object constraint into precedence constraints is carried out by distinguishing, for each object *o*, three kinds of tasks: *sources* (tasks which produce but do not require *o*), *sinks* (tasks which require but do not produce *o*), *pumps* (tasks which both require and produce *o*). Referring to the office example, tasks `"pick up document"` and `"take document"` are, respectively, a source and a sink for object `document`; `"photocopy document"` is a pump for `document` and a source for `copy`. In general, a task *s* is then translated as follows:

- If *s* is a source for *o*, its precedence set is left unchanged.
- If *s* is a sink for *o*, its precedence set is enlarged with all the sources and the pumps for *o*.
- If *s* is a pump for *o*, its precedence set is enlarged with all the sources for *o*.

When negotiating, the precedence constraints derived from object constraints cannot be attacked as simply as shown in Section 3.4. Consider an agent $A$ that has been assigned a couple of tasks $s_1$ and $s_2$ involved in the same object flow ($s_1$ produces one or more objects required from $s_2$). If one of the two tasks is negotiated and assigned to another agent $B$, it is necessary for $A$ and $B$ to meet, after $s_1$ has been carried out and before $s_2$ is started, so that the object(s) can be transferred from one agent to the other.

Some strategies which could be pursued to arrange the meeting between two self-interested agents are listed below:

1. *Propose-hence-pay*. The announcer commits itself to reach the awarded agent, that pays no additional cost.
2. *Require-hence-pay*. The agent taking the task which requires the object(s) commits itself to reach the other agent; thus, the agent taking the task which produces the object(s) pays no additional cost.
3. *Distribute*. The two agents agree on a meeting point placed halfway between the two places where $s_1$ and $s_2$ must be executed; thus, the cost for transferring the object(s) is distributed between the two agents.

Strategies (1) and (2), though perfectly reasonable from a cognitive point of view, fail in two cases: when the same task is exchanged twice or more times between different agents; and when the tasks which, respectively, produce and require a given object are exchanged with different agents. In both cases, in fact, these strategies may lead one or more agents to act as go-betweens, that is, taking and delivering objects they neither require nor produce.

Strategy (3) cancels this drawback; in fact, the meeting place is defined a priori by the announcer on the basis of the tasks involved in the object flow, independently of which agents will execute them (see Figure 2). The announcer determines an execution plan for its tasks by applying the path planning algorithm; this plan defines the flow of the objects through the tasks, i.e., for each object required from a given task, exactly one task which produces that object is univocally determined, and vice versa.

For each produce-require pair of tasks, as determined by the announcer, two *transfer tasks* are generated, both requiring the agent to visit the meeting place at the same time. One of the transfer tasks is a sink for the object and is associated to the producing task, the other is a source and is associated to the requiring task. Both the producing and the requiring tasks are always announced and bid together with their transfer tasks.

## 4 The Negotiation Strategies

The strategies which the contractors may adopt may be described as follows:

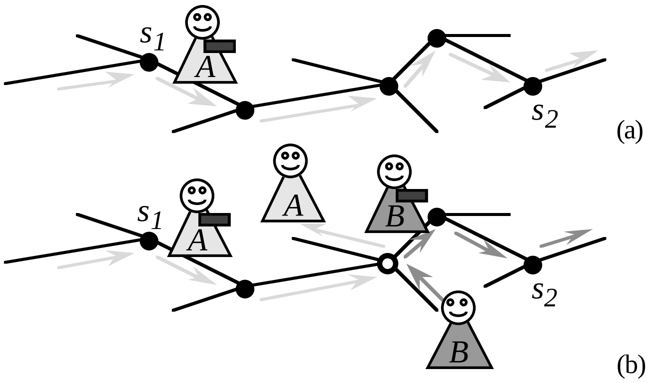


Figure 2. Exchange of a task involved in an object flow. (a) Path planned before the exchange; (b) Paths planned after the exchange. The small dark box represents object *o*; the white landmark is the meeting place, reached by both agents at time *t*.

- The announcer, in selecting the task to negotiate, cannot calculate the utility it will receive; thus, a reasonable heuristic it can follow is to choose the task $s*$ with the highest RMC. This heuristic can be improved by weighing the RMC of each task according to how far in the future that task has been scheduled; in fact, swapping a task *s* planned to be carried out at the end of the path may be hazardous, since the RMC of *s* may change substantially during the following negotiation sessions.
- The bidder can calculate the utilities it will get by swapping $s*$ with each of its tasks. After dropping all the non-individual rational swaps, it will decide how to formulate the bid. The most "politically correct" strategy it can adopt is to include in the bid all the individual rational swaps. This strategy maximizes the probability of success and the utility of the announcer; unfortunately, it may produce little utility for the bidder. Alternatively, the bidder can decide to include only the most convenient swap(s) in the bid. In this case the bidder is inclined to accept a reduction of the probability of success in exchange for a more advantageous deal.

Agent strategies are strictly related to protocol stability. The sale-based CNP as described in [Sandholm, 1993] is not stable. In fact, the announcing agent *A* associates each task announced with the maximum cost it is ready to pay for it, say $v_A$. Let *B* be a bidder that, for executing the announced task, would pay $v_B<v_A$. Knowing $v_A$, *B* can increase its utility to the detriment of *A*. In fact, *B* knows that, by bidding any value $v*$ such that $v_B \leq v* < v_A$, the contract will still be individual rational for *A*. As a matter of fact, *B* is stimulated to lie (but the higher its bid, the higher the probability that *A* will award the task to some other agent).

Remarkably, due to the absence of a common metrics to evaluate utility, the swap-based protocol has a higher stability than the sale-based one. In fact, the agent that artificially reduces its bid not only takes the risk that the announcer will award some other agent, but also that the swap it proposes will not be individual rational for *A*.

## 5 Experimental Results and Conclusion

In this paper we have proposed a negotiation protocol which extends the CNP by introducing task swapping, and we have discussed its application to path planning in autonomous robots. Different design alternatives have been considered, each characterized by different properties. In the alternative implemented, the announcement consists of a single task and the bid of a set of tasks.

In order to evaluate the performance of the protocol proposed, we have conducted simulations on a sample map including about 200 landmarks clustered on three layers. Each agent is assigned a set of 10 tasks on resources distributed uniformly over the environment.

Let $S_i$ be the set of the current tasks of agent $A_i$. We define the *saving* of the swap between the tasks $s_2 \in S_i$ and $s_1$ as

$$sSav\ (s_1, s_2, A_i, S_i) = \frac{uti\ (s_1, s_2, A_i, S_i)}{c_{A_i}(path\ (S_i))}$$

Let $S'_i$ be the set of the tasks assigned initially to agent $A_i$ and $S''_i$ be the set of the tasks actually executed by $A_i$. We define the *global saving* for the agent society $\{A_1,...A_k\}$ as:

$$gSav = \frac{\sum_{i=1}^{k} c_{A_i}\ (path\ (S'_i)) - \sum_{i=1}^{k} c_{A_i}\ (path\ (S''_i))}{\sum_{i=1}^{k} c_{A_i}\ (path\ (S'_i))}$$

Table II shows how the global saving and the average saving per swap per agent depends on the communication radius $\rho$ for a sample set of sessions: the higher $\rho$, the higher the number of successful swaps and their utility, hence, the higher the global saving.

Table III shows how the saving and the total number of successful swaps varies with the number of agents for a sample set of sessions: the more agents involved, the higher the probability of agreeing on individual rational swaps.

The experimental results obtained confirm that, by adopting swap-based contracts in the CNP, the costs for executing tasks in a society of self-interested agents can be effectively reduced. Our future work will focus on investigating how swap-based negotiations can be profitably coupled with sale-based ones in order to overcome local maxima in the utility functions of the agents.

## References

[Allen, 1991] James F. Allen. Planning as temporal reasoning. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, April 1991. Morgan Kaufmann.

| ρ | *avg. saving per swap per agent* | *global saving* |
|---|---|---|
| 1 | 0.07 | 0.34 |
| 1/2 | 0.04 | 0.19 |
| 1/4 | 0.03 | 0.17 |

Table II. Global saving and average saving per swap per agent in function of the communication radius for 4 agents (ρ is expressed as a fraction of the total diameter of the map).

| *n. of agents* | *avg. saving per swap per agent* | *max. saving* | *n. of swaps* | *global saving* |
|---|---|---|---|---|
| 2 | 0.03 | 0.05 | 4 | 0.12 |
| 3 | 0.03 | 0.09 | 9 | 0.17 |
| 4 | 0.04 | 0.17 | 10 | 0.19 |
| 8 | 0.05 | 0.21 | 23 | 0.29 |

Table III. Average saving per swap per agent, maximum saving achieved, total number of swaps and global saving in function of the number of agents (ρ=1/2).


[Causse and Crowley, 1994] Olivier Causse and James L. Crowley. Navigation with constraints for an autonomous mobile robot. *Robotics and Autonomous Systems*, 12(3-4):213-221, 1994.

[Dean *et al.*, 1988] Thomas Dean, James Firby and David Miller. Hierarchical planning involving deadlines, travel time and resources. *Computational Intelligence*, 4(4):381-398, 1988.

[Ghallab and Laborie, 1995] Malik Ghallab and Philippe Laborie. Planning with Sharable Resource Constraints. In *Proceedings International Joint Conference on Artificial Intelligence*, pages 1643-1649, Montreal, Canada, August 1995.

[Ghallab and Vidal, 1996] Malik Ghallab and Thierry Vidal. Dealing with uncertain durations in temporal constraint networks dedicated to planning. *CONSTRAINT-95: International Workshop on Constraint-Based Reasoning*, Melbourne Beach, USA, April 1995.

[Kakehi and Tokoro, 1993] Rumiko Kakehi and Mario Tokoro. A negotiation protocol for conflict resolution in multi-agent enviroments. In *Proceedings of International Conference On Intelligent and Cooperative Information Systems*, pages 185-196, May 1993.

[Lander and Lesser, 1993] Susan E. Lander and Victor R. Lesser. Understanding the role of negotiation in distributed search among heterogeneus agents. In *Proceedings International Joint Conference on Artificial Intelligence*, Chambery, France, August/September 1993.

[Maio and Rizzi, 1996] Dario Maio and Stefano Rizzi. Layered knowledge architecture for navigation-oriented environment representation. Technical Report CIOC-C.N.R., n. 108, 1996.

[Maio *et al.*, 1996] Dario Maio, Davide Maltoni and Stefano Rizzi. Dynamic Clustering Of Maps In Autonomous Agents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1080-1091, November 1996.

[Malone *et al.*, 1988] Thomas W. Malone, Richard E. Fikes, Kenneth R. Grant and Michael T. Howard. Enterprise: a market-like task scheduler for distributed computing environments. In *The Ecology of Computation*, pages 177-205. North-Holland, 1988.

[Rosenschein and Zlotkin, 1994] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter*. MIT Press, Cambridge, Massachusetts, 1994.

[Sandholm, 1993] Tuomas W. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings 11th National Conference on Artificial Intelligence*, pages 256-262, Washington D.C., July 1993.

[Sandholm and Lesser, 1995] Tuomas W. Sandholm and Victor R. Lesser. Issues in automated negotiation and electronic commerce: extending the contract net framework. In *Proceedings First International Conference on Multiagent Systems*, pages 328-335, San Francisco, California, June 1995.

[Sen, 1994] Sandip Sen. The role of commitment in cooperative negotiation. In *Proceedings International Journal on Intelligent and Cooperative Information Systems*, 3(1):67-81, 1994.

[Smith, 1980] Reid G. Smith. The Contract Net Protocol: high-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104-1113, December 1980.

[Timpf *et al.*, 1992] Sabine Timpf, Gary S. Volta, David Pollock and Max J. Egenhofer. A conceptual model of wayfinding using multiple levels of abstraction. In *Theories and methods of spatio-temporal reasoning in geographic space*, A.U. Frank, I. Campari and U. Formentini Eds., Lecture Notes in Computer Science, vol. 639, pages 348-367, 1992. Springer-Verlag.

[Wilkins, 1988] David E. Wilkins. *Practical Planning*. Morgan Kaufmann, San-Matteo, California, 1988.