

Research Article

A Heuristic Approach to Proposal-Based Negotiation: with Applications in Fashion Supply Chain Management

Stefania Costantini,¹ Giovanni De Gasperis,¹ Alessandro Provetti,^{2,3} and Panagiota Tsintza¹

¹ Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, 67100 L'Aquila, Italy

² Department of Mathematics and Informatics, University of Messina, 98166 Messina, Italy

³ Oxford-Man Institute of Quantitative Finance, University of Oxford, Oxford OX2 6ED, UK

Correspondence should be addressed to Alessandro Provetti; ale@unime.it

Received 14 November 2012; Accepted 16 November 2012

Academic Editor: Tsan-Ming Choi

Copyright © 2013 Stefania Costantini et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we extend and improve the formal, executable framework for automated multi-issue negotiation between two autonomous competitive software agents proposed by Cadoli. This model is based on the view of negotiation spaces (or “areas”), representing the admissible values of the goods involved in the process as convex regions. However, in order to speed up the negotiation process and guarantee convergence, there was the restriction of potential agreements to vertices included in the intersection of the two areas. We present and assess experimentally an extension to Cadoli's approach where, for both participating agents, interaction is no longer vertex based, or at least not necessarily so. This eliminates the asymmetry among parties and the limitation to polyhedral negotiation areas. The extension can be usefully integrated to Cadoli's framework, thus obtaining an enhanced algorithm that can be effective in many practical cases. We present and discuss a number of experiments, aimed at assessing how parameters influence the performance of the algorithm and how they relate to each other. We discuss the usefulness of the approach in relevant application fields, such as, for instance, supply chain management in the fashion industry, which is a field of growing importance in economy and e-commerce.

1. Introduction

Negotiation is a decision-making process in which multiple parties jointly make decisions to cope with conflicting interests. In particular, automated negotiation can be considered as a kind of interaction (i.e., a dialogue) in which some agents, with a desire to cooperate but with conflicting interests, work together with the aim of reaching a common goal [1, 2]. Nowadays negotiation, and particularly automated negotiation, deals with complex service features or products and is becoming increasingly important as a consequence of the rapid development of web-based transactions and e-commerce. There is no doubt that negotiation is an important process in many industrial processes and supply chains, and, as widely recognized in the literature (see the following), solutions based on autonomous intelligent software agents are gaining credit.

Negotiation can also be defined as “a distributed search in a space of potential agreements” [3]. Each participant involves

her/his individual area of interest (called the *negotiation space* or *area*), and intends to reach agreements (or “contracts”) in there. Negotiation spaces can be represented by a set of constraints, and finding an agreement can be modeled as a *constraint satisfaction problem* (CSP).

The particular mechanism of negotiation considered in this work is *proposal-based negotiation*. In this case, the process involves a number of agents which usually have a limited common knowledge about the other agents' constraints, that is, about the others' *negotiation spaces* (or “areas”). In proposal-based negotiation, information exchanged between the parties is in form of offers or proposals, that is, in the form of potential agreements that each agent proposes to the others (rather than in form of constraints, preferences, or argumentation). Proposals can be accepted or rejected and are possibly followed by counteroffers. Proposals are assumed to be internal points of the proponent agent's negotiation space; that is, each agent is “honest” and thus proposes only potential agreements that it is actually able/willing to accept.

Each agent is supposed to be able to compute the points of its own individual area and to use them as proposals which are communicated to the other parties. Preferences/priorities or other constraints in choosing these points are not considered (at least explicitly, though presumably they will have a role in selecting an offer), and the only possible actions are making, refusing, or accepting a proposal.

The research reported in this paper presents the extension (preliminary introduced in [4]) to a previous work by Cadoli in [5] and discusses a set of experiments. The approach by Cadoli, called Reasoning by Means of Projections (for short RMP), concerns proposal-based negotiation, from the point of view of minimizing the number of the interactions between the automated agents involved in the negotiation process. The point is to reduce the computational complexity and thus speed up the search of an agreement. In fact, this approach tries to accelerate the localization of agreements, where potential agreements are all common points of the agents' feasibility regions (i.e., points of the intersection area). Negotiation spaces are considered as convex: thus, all points between two potential agreements are acceptable as well. The admissible offers are in principle all the internal points of the negotiation areas, and those will be the only information exchanged between the agents. However, in order to speed up the negotiation process and guarantee convergence, there was the assumption/limitation that, for one (and only one) of the two agents, only vertices of its individual negotiation area are proposed as offers to the opponent. Therefore, only those vertices included in the intersection of the two areas constitute actual potential agreements. The key aspect of Cadoli's proposal is that agents are supposed to be able to reason in terms of geometric projections of their (convex) negotiation areas, as illustrated in the following. This kind of reasoning can help them to compute subsequent offers: in fact, as discussed later, agents are thus able to exclude certain points of the individual negotiation areas, so that the overall process is in the average case very efficient in terms of the number of steps required for the negotiation to converge, that is, for the parties to find an agreement.

The method developed in [5] is in contrast to existing approaches to negotiation that rely on a "mediator" agent, that is, on a third party authority who accepts and sorts out offers and proposes bids. The first one is probably the well-known "Contract Net Protocol" [6]. However, adopting such a solution forces the participating agents to disclose their policies and preferences. When privacy is important, such solutions cannot easily be adopted. There are also approaches that adopt a nonbiased mediator in the negotiation (see, e.g., [7]). Once they accept the mediator intervention, the negotiation agents do not have any right to make or accept offers based on their own negotiation strategies. This of course limits the application of these solutions.

In this paper, we present and assess experimentally an extension to Cadoli's approach (that we call ERMP for "Extended RMP") where interaction is no longer vertex-based, or at least not necessarily so. That is, we allow both agents to potentially make offers that are an internal point of its negotiation space and then try to approach the opponent's counterproposal "step by step." The extension presented here

overcomes some problems of the original one, such as the asymmetry among the parties (only one agent is allowed to use RMP otherwise problematic situations occur), the "flat" nature of RMP, where no agents' preferences and utilities are considered (this aspect is also considered in [8]), and the limitation to polyhedral negotiation areas. Our extension can be usefully integrated into Cadoli's framework, thus obtaining a new algorithm that may be effective in many practical cases by introducing *local search*, for instance, around, the "best-preferred" vertices.

ERMP has a potential practical applicability in many areas related to e-commerce or supply chain management (SCM), where there is a growing interest in autonomous interacting software agents and their potential application to support actual negotiation and contract making. For instance, negotiation within strict time-to-market constraints is a crucial component of the supply chain in the fashion industry, a field which has a growing importance in economy in general and in e-commerce in particular. Agent-based or agent-supported negotiation can help overworked human managers, where intelligent agents can be equipped with their own strategies and objectives. Decision makers being represented by agents can help to make better decisions in a shorter time, as argued, for example, in [9] and, specifically for fashion business, in [10].

This paper is structured as follows. Section 2 gives an overview of the theoretical background and discusses the motivations of this work. In Section 3 we present the proposed extension and the algorithm that we have adopted for the implementation. We also provide necessary conditions for convergence and propose an integration with the original approach. Section 4 briefly reports about the implementation and shows a snapshot of the code. The next section is thus dedicated to the experiments that we have performed and to explain the results of the testing phase. Our experimental assessment, which has involved a large number of instances, seeks to determine which parameters influence the performance of the algorithm, and in which way, and how the different parameters are related to each other. We have also tried to assess whether the algorithm is reliable, in the sense that it does not show unwanted or unexpected behavior. In Section 4.1 we compare our strategy to the original one from the point of view of performance/scalability. A discussion, with comparison to the current literature and some insights of potential applications (especially in the fashion business field), concludes this paper.

2. Reasoning by Means of Projections (RMP)

In this section we first present the approach by Cadoli [5] (called RMP, for short) and then discuss why some extensions are needed and useful. In RMP, negotiation is modeled as a distributed constraint satisfaction problem whose features are illustrated in the following. In his work, Cadoli made several assumptions about the form of negotiation; let us discuss them now.

- (i) Negotiation involves two or more parties that exchange proposals until either an agreement is found,

that is, the last proposal is acceptable by all the parties involved, or there is an evidence of the fact that no agreement is possible. Without loss of generality it is going to be assumed that negotiation involves only two parties, that later on will be called “agents.”

- (ii) Negotiation involves variables (also called negotiation issues). A proposal (or “offer”) consists in communicating to the other party a possible assignment of values to the involved variables (each one is called a “variable assignment”).
- (iii) Negotiation is restricted without loss of generality to involve only two variables. The approach however could be easily generalized.
- (iv) The negotiation space (also called negotiation region or feasibility region) associated with each party coincides with the set of variable assignments that are considered to be acceptable, that is, where the value assigned to each variable is within the range that the party considers to be acceptable. Variables are considered to denote real numbers.
- (v) As only two variables are involved, negotiation spaces are restricted to be regions in the Cartesian plane.
- (vi) A possible proposal is, in principle, any point of the negotiation space.
- (vii) Negotiation spaces are restricted to be convex regions; that is, all points (pairs of real numbers) included within the boundaries of each individual region belong to the region itself and thus are equally acceptable as potential agreements. Therefore, each negotiation space can be described by a set of constraints which describe the region perimeter by describing the acceptable range of values for each variable.
- (viii) Negotiation spaces are considered to be polyhedral. Thus, negotiation spaces admit a finite number of vertices.
- (ix) Possible proposals are asymmetrical for the two agents. One of them is supposed to have the objective of minimizing the number of iterations, and in this perspective it offers only the vertices of negotiation spaces. The other one can instead offer any point of the negotiation space. The reasons of this asymmetry will be discussed in the following.

Of course, there are other assumptions that determine the negotiation scenario. For instance, it is assumed (again without loss of generality) that the two parties have agreed in advance on the issues which are involved. Also, parties associate a meaning to the variables, that in their view may represent prices, time, number of items, and so forth.

A negotiation starts when one of the two agents makes a proposal. The other one will respond with a counterproposal, and the process will go on in subsequent *steps* (where each agent responds to the other one’s last proposal). The other party is also called the “opponent.” As mentioned, the negotiation process will end either because an agreement has been found or if there is an evidence that no agreement is possible.

Since negotiation spaces are considered as convex regions, a necessary condition for an agreement to be reached is that the intersection of the feasibility regions is not empty.

As offers are restricted for one of the two agents to be vertices of a polyhedral region, in the RMP approach the process will necessarily end whenever this agent has no more vertices to offer. This means that each negotiation process always converges to an end in a finite number of steps. However, the number of vertices can be, in the worst case, exponential in the number of variables. Thus, the process has an exponential worst-case complexity in the number of negotiation steps. Therefore, the approach defines a negotiation protocol aimed at obtaining on average large savings in terms of number of proposed vertices. The underlying assumptions about the participating agents are the following.

- (i) Agents are perfect rational reasoners.
- (ii) Agents communicate only by exchanging proposals. There is no other form of shared knowledge. When a proposal has been issued, it becomes common knowledge for all involved parties.
- (iii) Agents are partially cooperative, in the sense that they are aware of the negotiation protocol that they apply faithfully, that is:
 - (a) they do not make offers that they are not really willing to accept;
 - (b) they do not cheat; that is, they do not make proposals that are not implied by the protocol at that step.
- (iv) Agents are able to reason by means of projections.

Let us now discuss the nature of *reasoning by means of projections* that RMP exploits. Regardless of its standard mathematical definition, *projection* can be defined intuitively as the image of a geometric figure on a line, plane, or surface. The projections of line segments over each other are then reasoned about. The projection of a vector b in the direction of a vector a is given by $\text{proj}_a b = (a \cdot b / |a|^2) a$.

So, the projection of b in direction a is a scalar multiple of it. The length of the projected vector is called the component of b in direction a : $\text{comp}_a b = ab / |a|$. This is designed so that the projection of b in direction a forms a right triangle with vector b , with the right angle at the end of the projected vector. This concept is recalled by means of an example, in Figure 1.

In the example, there are considered the four points A_1 , A_2 , B_1 , and B_2 . The figure highlights two projections:

- (1) the projection of the segment $\overline{A_1 A_2}$ over $\overline{B_1 B_2}$ (denoted with $\Pi(\overline{B_1 B_2}, \overline{A_1 A_2})$ and delimited by $r_2 \overline{B_1 B_2} r_1$),
- (2) the projection of $\overline{B_1 B_2}$ over $\overline{A_1 A_2}$ (denoted with $\Pi(\overline{A_1 A_2}, \overline{B_1 B_2})$ and delimited by $s_1 \overline{A_1 A_2} s_2$).

Suppose, for instance, referring to Figure 1, that points A_1 and A_2 are offers made by agent A , while B_1 and B_2 are offers made by agent B . Assume that A is the party who, in order to minimize the number of iterations, is bound to offer vertices

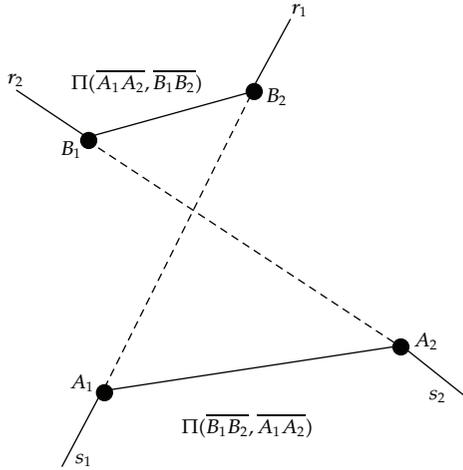


FIGURE 1: Line segments' projections.

of its negotiation area, while instead B is allowed to offer any point.

To exemplify it, assume the following exchange: agent A offers A_1 , agent B counteroffers B_1 , and agent A replies with A_2 and agent B with B_2 . At such point, agent A (being a perfect, logical reasoner and being aware of convexity and projections and being aware that the other agent has the same potential) is able to perform the following reasoning.

From proposals A_1 and A_2 and from the convexity hypothesis, agent B knows that the whole segment $\overline{A_1A_2}$ belongs to A 's negotiation area. Then, as B aims at minimizing the number of steps, if B had a point of its region on that segment, it would have offered it. As B instead has offered B_2 , agent A is allowed to conclude that the intersection between B 's region and segment $\overline{A_1A_2}$ is empty.

Also, A can consider that if B had any vertex on either the line s_1 that goes beyond segment $\overline{B_2A_1}$ in the direction of A_1 or on the line s_2 that goes beyond segment $\overline{B_1A_2}$ in the direction of A_2 then, again by convexity, the segment $\overline{A_1A_2}$ would necessarily belong to B 's region: in fact, any polyhedral region including as vertices B_1 and B_2 and these two hypothetical points would also include segment $\overline{A_1A_2}$, which is a contradiction. Therefore, A is able to exclude the whole projection area delimited by segment $\overline{A_1A_2}$ and by the two above-mentioned lines. In fact, no point in there can be possibly acceptable for B , and thus A will choose no such point as an offer. This area can be obtained, as observed before, by making the projection of segment $\overline{B_1B_2}$ over $\overline{A_1A_2}$.

Excluding portions of the interaction area leads to excluding many potential offers and thus reducing the number of steps. In fact, agent B can perform a similar reasoning if A responds to offer B_2 with a counteroffer that does not constitute an agreement. The reasoning can then be repeated by both agents after subsequent offers.

Notice that this example shows that the asymmetry in the nature of the offers from the two parties is an essential part of the approach. In fact, agent B is able to offer a point on segment $\overline{A_1A_2}$ because it is able to offer any point of the

negotiation area. If instead B could offer only vertices, then it would be able to offer a point on segment $\overline{A_1A_2}$ only if it had a vertex there.

If both parties are aimed at minimizing the number of steps and thus use the same protocol and offer only vertices, possible agreements might be excluded. The agreements might fail to be reached also for other reasons. This is one of the reasons leading to the extension of this approach. In summary, the interactions among negotiating agents involve the following factors.

- (i) *Variables*: it is assumed that, before starting a negotiation, the agents agree on the number and on the nature of the variables considered during the process. Variables represent issues that the agents wish to take into account and are numeric (or Boolean as special cases of integers). In the considered examples all variables denote real numbers.
- (ii) *Constraints*: the negotiation areas of each agent can be represented as a set of constraints. A proposal can be accepted only if it is included on the individual negotiation area (that means that the set of constraints is satisfied). Note that the individual negotiation area (i.e., the set of all the admissible offers) is private knowledge of each agent. Also the projection areas computed by the agents are represented by sets of constraints. The constraints are private knowledge of each agent.
- (iii) *Information exchanged*: it is assumed that the negotiation adopts a *proposal-based* mechanism. This means that the only information exchanged among agents consists of offers and counteroffers. At least one agent is bound to select as offers only the area vertices. Thus, this approach is applicable only in case of polyhedral negotiation areas. All other information items, such as constraints or preferences, will be considered as private. The agent's response to each offer can be either acceptance, that concludes the process, or rejection, that induces a counteroffer (if existing).
- (iv) *Protocols and agents' cooperation level*: the protocols can be considered as the rules that each party involved in the process of negotiation has to respect. In this context, the agents are bound to communicate only the offers included in their individual areas and that they are really willing to accept. Moreover, agents are *logically omniscient*, which implies that each of them is able to draw conclusions from what other agents are assumed to know. In this sense, the agents are considered to be *partially cooperative*.
- (v) *Strategy*: this approach adopts the strategy of concluding the process in the minimum possible number of interactions. This means that the approach tries to find an offer included in the common part of the negotiation areas by performing the least possible number of steps.
- (vi) *Objective function*: each agent is self-interested, which means that it ignores the other agents' preferences.

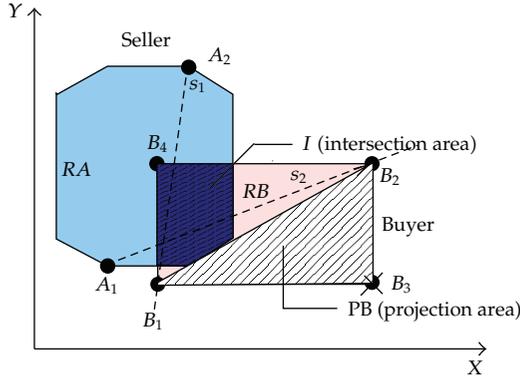


FIGURE 2: An example of negotiation.

Notice that, in this setting, finding an agreement corresponds to solving a *distributed constraint satisfaction problem* (DCSP) where initially only the negotiation areas are considered and subsequently extended by the projection areas.

Let us now consider a more complex (and complete) example. Consider agents *Seller* and *Buyer*, who are involved in the negotiation process that is represented by the respective negotiation areas in Figure 2. Moreover, assume that agent *Buyer* must offer vertices while the other agent simply *prefers* to offer vertices and thus resorts to internal points only when an agreement would not be otherwise possible.

In this example, the negotiation area (indicated as R_A) of agent *Seller* can be described by the set of constraints

$$C_A = \left\{ \begin{array}{l} X \geq 4, X \leq 20, Y \geq 13, Y \leq 40, \\ X \geq -3Y + 49, X \leq Y + 4, \\ X \geq \frac{3}{2}Y - 50, X \leq -\frac{3}{4}Y + 47 \end{array} \right\}. \quad (1)$$

The negotiation area (denoted as R_B) of the agent *Buyer* has been described by the set

$$C_B = \{X \geq 15, X \leq 40, Y \geq 10, Y \leq 25\}. \quad (2)$$

Now, let

$$V_A = \{A_1 = (10, 13), A_2 = (17, 40), \\ A_3 = (4, 15), A_4 = (17, 13), \\ A_5 = (20, 16), A_6 = (20, 36), \\ A_7 = (4, 36), A_8 = (10, 40)\} \quad (3)$$

be the sets of possible proposals (set of vertices of the negotiation space, obtained by resolving the *Seller's* set of constraints) of agent *Seller*, and let $V_B = \{B_1 = (15, 10), B_2 = (40, 25), B_3 = (40, 10), B_4 = (15, 25)\}$ be the possible proposals of agent *Buyer*. The intersection area $I = R_A \cap R_B$ is clearly not empty, and therefore there is a potential agreement between the two agents.

Assume that the negotiation process starts with a proposal from the agent *Buyer* and that the sequence of proposals is

as follows: B_1, A_1, B_2, A_2 . Each interaction has the side effect of updating the knowledge base of the agents by storing all proposals (either made and received) and possible new constraints. The subsequent steps of the negotiation are determined as follows.

- (i) Agent *Seller* refuses the offer B_1 and continues the negotiation dialogue by proposing its vertex A_1 .
- (ii) Since agent *Buyer* has received as counteroffer point A_1 , not included in its negotiation area, it rejects the proposal and offers its vertex B_2 .
- (iii) Also this new proposal is not accepted from the agent *Seller* which responds with its vertex A_2 .
- (iv) After two interactions, the agents have exchanged four proposals, namely, B_1, A_1, B_2 , and A_2 , and none of them has been accepted. At this point, agent *Buyer* computes a projection area by connecting the couples of points (B_1, A_2) , (B_2, A_1) , and (B_1, B_2) and by adding to its knowledge base the new linear constraints that represent the new lines s_1 and s_2 . Agent *Buyer* thus concludes that the negotiation area of agent *Seller* has no common point with the infinite area (called the *projection area*) delimited by lines s_1 and s_2 and by the segment $\overline{B_1B_2}$. The proof of this conclusion is as follows.

- (1) The two proposals B_1 and B_2 imply (from the convexity hypothesis) that agent *Seller* knows that $R_B \supseteq \overline{B_1B_2}$.
- (2) Since agent *Seller* has proposed its vertex A_2 , agent *Buyer* knows that $R_A \cap \overline{B_1B_2} = \emptyset$.
- (3) Agent *Buyer* is now able to conclude that its projection area has no common point with agent *Seller's* negotiation area. This is because if there were such a point, then the entire triangle $\overline{B_1PB_2}$ would have to be included—from the convexity hypothesis—in the R_A area. As this triangle has at least one point in common with segment $\overline{B_1B_2}$, a contradiction arises.

Agent *Buyer* will exploit the updated knowledge so as to select the next offer to make, excluding its vertex B_3 that cannot be accepted by the counterpart. Thus it proceeds to offer vertex B_4 .

- (v) Finally, this last offer belongs to the negotiation area of the agent *Seller*, and therefore this proposal will be accepted. In this case, the negotiation process terminated successfully.

Notice that the generalization of the approach to n variables is, from a mathematical point of view, straightforward. We do not comment on this aspect here for the sake of conciseness, because the explanation would necessarily be lengthy. We may notice, after [5], that the number of vertices of negotiation areas grows exponentially with the number of variables, so the advantage of the method is even greater. For instance, in fact, an area defined by the set of $2n$ constraints $0 \leq x_i$;

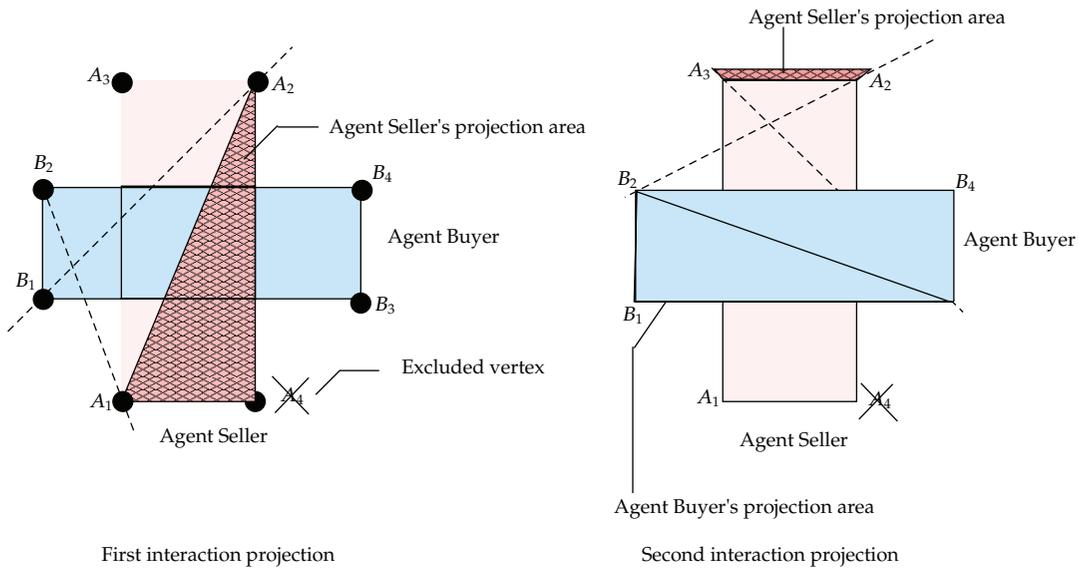


FIGURE 3: A first problematic interaction.

$x_i \leq 1$ ($1 \leq i \leq n$) on the n variables x_1, \dots, x_i describes a hypercube with 2^n vertices. Reasoning by projection allows to obtain large savings in terms of number of proposed vertices, thus making multi-issue negotiation more effective.

There are however other ways to cope with a larger number of variables while staying within the two-issue case discussed before. One of the two issues might be the most relevant variable (e.g., price or time of delivery), and the other issue might represent a combination of the other variables, possibly according to some utility function (the reader may refer, e.g., to [11] for an interesting discussion of utility functions). Another solution is sequential negotiation, that is, when two agents negotiate issue by issue sequentially, establishing initially the order to negotiate each issue. A completely different way of coping with multi-issue negotiation and utility functions is based on exchanging (as offers) constraints instead of variable values (see, e.g., [12]). Other solutions are widely discussed in [13].

2.1. Limits of the Approach. Some proposed extensions to the basic RMP approach are motivated by the following observations.

- (i) Limiting the possible proposals to vertices is efficient but has some limits. One of them is that only one agent can adopt this strategy, that in fact works only if the other agent is instead capable of offering any point of the feasibility region. Actually, RMP makes the implicit assumption that the parties have agreed in advance of the respective strategies. If both agents instead offer only vertices, in some cases this induces problematic trade-offs: in particular, whenever the intersection area is not empty but includes no vertices. For example, in Figure 3 there are two agents, *Seller* and *Buyer*, whose individual negotiation areas are expressed through convex regions.

In this situation, there is clearly a potential agreement amongst the agents since the intersection area is nonempty. However in this case, after six interactions (namely, the sequence of proposals is $B_1, A_1, B_2, A_2, B_3, A_3$) the seller agent understands that there is no other vertex to propose: in fact, it has previously excluded vertex A_4 . Thus, it concludes the process of negotiation with a proof that there is no possible agreement. This problem is due to the fact that the intersection area, though not empty, does not contain any vertex.

- (ii) The *flat* nature of proposals (where all vertices are equally considered) may lead, in real applications, to other problematic situations. In particular, as agents may have their (private) preferences and objectives, it may frequently happen that one or more issues have for an agent greater priority than others. That is, the agent *prefers* to maximize/minimize this issue rather than others. This accounts to assuming that the agents try to reach an optimal point with respect to a private objective function. This objective function can be chosen according to the particular context. In Figure 4, for instance, there are considered two agents, *Business* and *Client*. Another assumption is that *Business* wishes to maximize the issue Y and that the sequence of interactions starts with a proposal (from *Client*) of point B_1 . The interaction proceeds with A_1, B_2 , and A_2 . In this case, the approach of reasoning by means of projections does not allow us to obtain savings in terms of number of proposed vertices.
- (iii) As mentioned in [5], another problem of the approach of reasoning by means of projections is that, since the agents have to remember all proposals made and received, it is hard to find algorithms and data

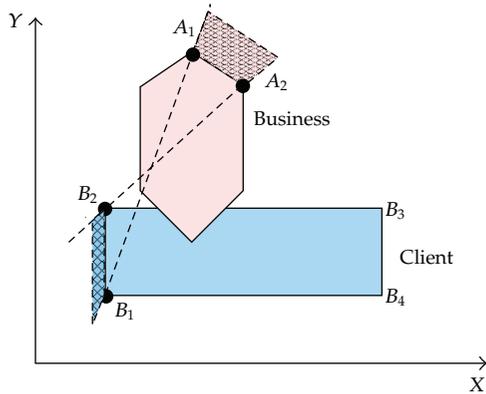


FIGURE 4: A second problematic interaction.

structures which allow agents to store the entire sequence of proposals in polynomial space.

- (iv) In [5], yet another problem is mentioned: the one of the worst case in which the negotiation spaces are convex (and finite) regions represented by circles. In this case, agents are unable to select their offers since there are no vertices to be selected.

Mancini [8] proposes an implementation and some relevant improvements to the approach: on the one hand, heuristics are introduced for offering the *best* vertices in terms of the efficiency of the overall process; on the other hand, negotiating agents are allowed to have their own (private) utility function to maximize, though defined so as to still guarantee convergence. In particular, in [8] agents can to some extent *lie*, by rejecting acceptable offers, hoping to increase their utility. The main limit of the RMP approach, that is, that only one agent can adopt the vertex-based strategy, is however not tackled.

The aim of the present work is that of coping with some of the problematic aspects and extending the basic approach under an important respect. In particular, proposals are assumed to be not only vertices but also internal points of the convex regions. However, since an infinite number of possible offers exists those points cannot be randomly selected. Rather, a protocol that carefully chooses the offers and keeps the advantages of reasoning by means of projections is proposed. Then, the introduced algorithm is more flexible: in fact, both agents are allowed to follow the same protocol, thus relaxing the main limitation of RMP. However, as we will demonstrate by means of extensive experiments, it is still reasonably efficient. We also intend to cope with the fact that the interest of agents in a contract usually declines gradually with distance from their ideal contract. Thus, we will introduce a form of “local search” starting from preferred vertices.

In the new algorithm, the next offer selection is based on recent proposals by the same agent, which will be increased (or decreased) by a δ margin. This margin can be chosen in various ways and will be in general application dependent. Criteria for such a choice are discussed, for example, (referring to an analogous setting) in [14]. Within the margin, the

next offer is selected so as to try to approach the opponent’s last offer.

3. A New Approach: ERMP

This section is dedicated to the introduction of a formal and executable extended approach to automated multi-issue negotiation between two partially cooperative agents. The proposed extension to the original Cadoli’s approach tries to respect its *spirit*, that is, that of improving average-case complexity by means of heuristics, and to keep the strong points, in particular by exploiting the reasoning by means of projections. Our aim is that of overcoming (at least some of) the problems identified in the original formulation and mentioned in the previous section. As it will be described in detail later, the new approach, called ERMP for “Extended RMP,” can be integrated to RMP thus obtaining a hybrid approach that can be useful in many practical cases.

Firstly, the set of agents involved in the process is defined as $AS = \{A, B, \dots\}$. Similar to the basic approach, for the sake of simplicity and without loss of generality the extension is illustrated by considering only two agents, called *A* and *B*. Then, the assumption that the agents have already agreed on the number and on the nature of the variables on which they negotiate persists. Let us suppose for instance, that the two parties have agreed upon using two variables denoting real numbers. In this way, the two negotiation areas can be represented in the Cartesian plane. The individual areas will be denoted as R_A and R_B . In general, as discussed previously the two bidding agents can reach an agreement only in the case where $R_A \cap R_B \neq \emptyset$.

The extension to [5] is based on a change of the nature of the proposals, which are allowed to be internal points of the feasibility areas rather than just vertices. The new algorithm still takes advantage of reasoning by means of projection. As before, at each stage of the process the bidding agents will never make proposals that are included in the projection areas identified so far, since those particular points will never be accepted by the antagonist. Similar to the basic approach, computing projection areas will lead to a great saving in terms of offers as parts of the individual negotiation areas are excluded.

In the extended algorithm the number of offers in the worst case (no agreement possible) is infinite. Nevertheless, the large number of experiments performed leads to the confidence about the fact that the proposed strategy speeds up the negotiation process whenever an agreement exists.

3.1. Identifying Proposals in the ERMP Approach. The approach proposed is based upon relaxing the limitation of offers to be vertices. Thus, it can be applied to any kind of convex negotiation areas, without restriction to polyhedral ones. But, considering that each negotiation area contains a huge (infinite) number of possible proposals, in which way and under which criteria are those points to be selected?

Here an example is proposed in order to explain the crucial point of the algorithm, that is, the choice-of-proposals phase. Assume that there are two agents (*A* and *B*) involved in the negotiation process. All possible offers of each agent

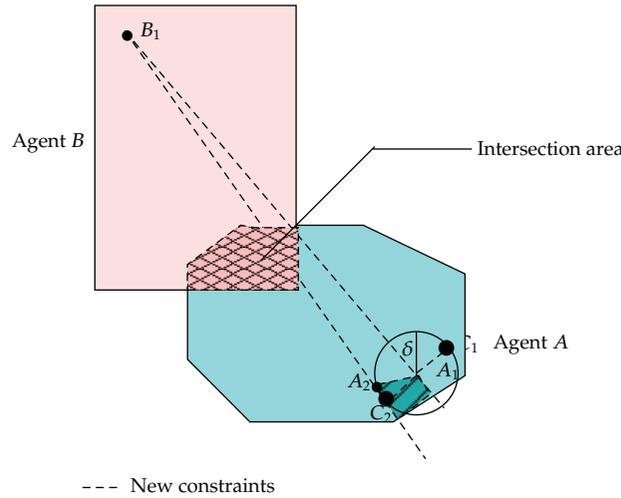


FIGURE 5: The trade-off strategy in multi-variable regions.

are represented in the Cartesian space as convex feasibility regions. The negotiation areas— R_A for agent A and R_B for agent B—as well as the necessary computation aimed at identifying the next proposal of both agents are reported in Figure 5. All remaining factors involved in the process are assumed to be identical to those considered in the original approach and fixed in advance. Let us assume that the negotiation process initiates with a proposal, for example, by agent A.

- (i) *First proposal*: choose a random point of the individual convex region of the agent. In Figure 5 the first two proposals are denoted as A_1 for agent A and B_1 for agent B.
- (ii) *Second proposal*: the agent computes the circumference centered on the point corresponding to its own first proposal (A_1 for agent A); the radius is set to $R = \delta$, where the choice of the margin δ will depend either on each specific application context or on the individual negotiation space. δ is set before starting the negotiation and is not modified at runtime. The next proposal will be selected as a random point of the semicircumference “closer” to the opponent’s proposal.

To do so, agent A acts as follows. To find the points C_1 and C_2 (Figure 5) delimiting the semicircumference

- (1) it computes the vector $\overline{C_1C_2}$ orthogonal to the vector $\overline{A_1B_1}$ (denoted by $\overline{C_1C_2} \perp \overline{A_1B_1}$), by setting their inner product to 0 ($\langle \overline{C_1C_2}, \overline{A_1B_1} \rangle = 0$),
- (2) it models and resolves the system (set) of equations composed of one of the inner products (4) and the equation of the circle centered in $A_1 = (A_{x_1}, A_{y_1})$ and radius δ (5). This means that, in the example, the system of equations will be the following. With $C_i = (Cx_i, Cy_i)$ and $i = 1, 2$,

(5) can be generalized to $(Cx_i - Ax_{i1})^2 + (Cy_i - Ay_{i1})^2 = \delta^2$:

$$\langle \overline{C_1C_2}, \overline{A_1B_1} \rangle = 0, \quad (4)$$

$$(C_1 - A_1)^2 + (C_2 - A_1)^2 = \delta^2. \quad (5)$$

- (3) Finally, by merging (4) and (5) it obtains two points of the circumference, namely, C_1 and C_2 . Now, agent A is able to split the circumference into two parts and choose the one that minimizes the distance (the “closer” one) to point B_1 (i.e., the first proposal of agent B). A’s next proposal will be, in fact, selected as a random point of this semicircumference (e.g., point A_2 in Figure 5). Clearly, this point is required to be included in the individual feasibility region.

In this way, each agent tries to approach the counterpart’s offer by proposing a point that is more likely to be accepted and by adapting the individual profile to the one that can be assumed for the opponent. To do this, the agent has to add new constraints to its knowledge base. If no such point is found, that is, in the case where the entire semicircumference is not included in the negotiation area, then the next proposal will be a new random point of the feasibility region.

After that, each agent exploits the reasoning on projection in the same way as in [5]. The projection area will be stored in the agent memory, by adding the new constraints that represent this area to its knowledge base. In such a way, the agent will be able to exclude all those points that will never be accepted by the opponent.

- (iii) *Subsequent proposals*: All subsequent proposals will be selected in the following way. Each agent computes the segment that connects the last two offers made by the opponent. Since the negotiation areas are convex,

the entire segment will be included in the opponent's region. Therefore, if the agent's area includes a point that belongs to this segment, then it corresponds to an agreement and thus will be the next offer to make.

Otherwise, subsequent proposals will be selected in the same way as the second one (where the center of new circumference will be the agent's last proposal) by adding, however, one further condition: the new proposals must not be included in the areas of the projections made so far. The projection areas may be described in terms of a set of linear equations. In this way, the agent will find the new variable values to offer by working out a new, or extended, DCSP.

An advantage of the proposed algorithm is that the agent does not have to store all past proposals. Rather, the only information that the agent needs in order to construct the new projections consists of the two most recent proposals made by each party. If an agreement has not been found yet, the agent continues with the next proposal and so on.

3.2. How to Guarantee Convergence. Since there is a huge (infinite, in principle) number of points included in an agent feasibility region, convergence of the extended algorithm is not guaranteed. To overcome this problem, two solutions are *prima facie* available.

A first simple empirical solution can be to introduce an upper bound to the number of allowed interactions. The upper bound is strictly needed in case there is no possible agreement: here in fact, subsequent attempts would otherwise go on indefinitely.

A second possible solution relies on an assumption about the size of the margin δ with respect to the size of the intersection area of the agents feasibility regions, whenever it is nonempty. In particular, it is assumed that the margin δ is smaller than half of the size of the intersection area for each of its dimensions: that is, in the Cartesian plane, δ is assumed to be less than a half of both the maximum width and the maximum height of the intersection area. If so, in order to guarantee convergence an agent can impose a *grid* on its feasibility region: on the Cartesian plane, this grid would consist of a set of parallel horizontal lines and a set of parallel vertical lines where the distance between two parallel lines is exactly δ (these lines can be defined by means of constraints that must be added to the others). The intersections among the lines define a finite set of points, called the *nodes* of the grid and which are uniformly distributed in the feasibility area. Under the given assumption on the size of δ , the intersection area contains some nodes. The algorithm should then be modified so as to always offer nodes. That is, when the agents, to select the next proposal, take a random point of the semicircumference "closer" to the opponent's proposal, it should then offer the node which is closest to this point. This guarantees that an agreement (if any) is always reached, and in any case the process terminates when there are no more nodes to offer.

3.3. Integration with the RMP Approach. ERMP can be usefully integrated with the original RMP approach. Assume

in fact that an agent has private preferences and objectives. This may often result in the fact that some vertices of the feasibility region are preferred to the others and that the part of the region nearby these vertices is preferred to the rest, in the sense that the agent would prefer to reach an agreement there.

An integration between the two approaches might allow the agent to still try to minimize the number of iterations, while also trying to fulfill its objective of reaching a "preferred" agreement. A possible integration is, for example, the following:

- (1) the agent might initially offer the preferred vertices;
- (2) if no agreement has been found, it might start ERMP not from a random point, but from a preferred vertex;
- (3) If step 2 has been repeated for all preferred vertices with no agreement, the remaining (nonpreferred) vertex will be offered;
- (4) if there are no more vertices to offer, the agent may optionally (as a last chance) start ERMP from a random point.

In this case, it looks reasonable to stop EMRT after a predefined number of steps. This integration captures the important aspect (emphasized, e.g., in [15]) that agents utilities for a contract usually decline gradually, rather than stepwise, with distance from their ideal contract.

This integration can be considered to be a "Local Search variant" of RMP. In fact, local search techniques are a family of general-purpose techniques for the solution of optimization problems [16]. Consider an optimization problem, and let S be a possible search space for it. A function N , which depends on the structure of the specific problem, assigns to each feasible solution $s \in S$ its neighborhood $N(s) \subseteq S$. Each solution $s_1 \in N(s)$ is called a "neighbor" of s . A local search technique, starting from an initial solution s_0 (which can be obtained with some other technique or generated at random), enters in a loop that navigates part of the search space, stepping iteratively from one solution to one of its neighbors. The modification that transforms a solution into one of its neighbors is called a move. The selection of the move to be performed at each step of the search is based on the cost function f , which assesses the quality of the solution. In our case, S is the feasibility region, s_0 is chosen randomly, the function N consists in the construction of the circumference with radius δ , and f selects a random point on this circumference.

4. DALI Implementation and Experimental Evaluation

This section describes our pilot implementation of ERMP by mapping it into the DALI language for multiagent systems (MAS) modeling and implementation. DALI [17–21] is an Active Logic Programming language designed in the line of [22] for executable specification of logical agents. DALI is a Prolog-like logic programming language with a Prolog-like declarative and procedural semantics [23]. The proposed

approach is decided to be implemented in the DALI language for the following reasons. DALI is a logic programming language and thus guarantees fast prototyping and a readable code and is suitable for modeling reasoning processes. DALI is agent-oriented and provides advanced capabilities of autonomy, reactivity, proactivity, and social abilities. This has allowed us to easily and directly model the main steps of the algorithm: in fact, the interaction requires social ability; initiating a negotiation is a proactive initiative of one of the parties; responding to a proposal involves reactivity and reasoning to come up with a counterproposal.

To support this claim, a snapshot of the code is presented, namely, the part that exploits the reactive capability of DALI agents. In order to improve elaboration tolerance of the implementation, the private knowledge of each agent participating in a negotiation (such as the constraints and the size of the δ parameter) is stored in the agent's profile and is loaded at runtime. The reactive rule used for the exchange of proposals has this form:

```
OfferE(X,Y,A) :> once(reconsider(X,Y,A)).
```

The connective `>` indicates that this is a reactive rule, which performs “forward” reasoning. In fact, `atom offerE(X,Y,A)` is an *external event*, as syntactically indicated with postfix *E*. This particular external event represents a proposal, where (X,Y) is the proposed point and *A* the proposer agent. Whenever such an event is received by the agent, the reactive clause “fires” and the body is executed, thus taking into consideration the proposal itself.

If the received proposal is included in the negotiation area of the agent (i.e., `in_area(X,Y)` holds), then it has to be accepted. This is encoded by the following rule, which in fact sends to *A* a message of acceptance:

```
reconsider(X,Y,A) : - in \_area(X,Y),!
messageA(clientnew, send_message
(accept_pr(X,Y,A,A)).
```

If instead the received proposal is not included in the negotiation area, then it is not accepted. Abstracting away from the details of the code, with the following rule the agent computes a counterproposal according to the above-illustrated algorithm, sends this counterproposal to the opponent, and updates its constraint knowledge base:

```
reconsider(X,Y,A) : - out_of_area(X,Y),
(Az > X, Bz > Y, out(→,→,→,→, X,Y)),
call_random_semicycle(X1,Y1,X,Y),
messageA(clientnew, send_message
(new_offer(X1,Y1,A,A)),
update_offer1(offer1(→,→), Ac,Bc),
update_offer2(offer2(→,→), X,Y),
update_proposal1(proposal1(→,→), Az,Bz),
update_proposta2(proposta2(→,→), X1,Y1),
clause (offers(L),→),
append([Y],L,L1), append([X],L1,L2), assert
(offers(L2)),
```

```
retractall(offers(L)),clause (proposals
(Lp),→),append ([Y1], Lp, L3),
append([X1],L3,L4), assert(proposals(L4)),
retractall(proposals(Lp)).
```

More precisely, `call_random_semicycle(X1,Y1,X,Y)` is the procedure that implements the search of a random point in the semicircle closer to the last offer proposed by the opponent, that is, the X, Y variables contained in the external event `offerE(X,Y)`. After that, the agent performs the action

```
messageA(clientnew, send_message
(new_offer(X1,Y1,A,A))
```

that sends a message containing the new point $(X1,Y1)$.

We have performed a number of experiments in order to analyze and validate the behavior of ERMP. In the experiments, that have involved a great number of instances, it is tried to be established which parameters influence the performance of the algorithm and in which way. It is also intended to establish different parameters that are related to each other.

Experiments have been performed on a real-world case study involving variables with a clear intuitive meaning. The case study is in the context of the fruit market. In this experiment, two agents, *Seller* and *Buyer*, involved in a fruit market are considered. The negotiating agents agree on taking the price of the product (let us say, apples) and the quantity (in pounds) as the negotiation parameters.

Assume that agent *Seller* is interested to sell the product for a minimum price of 1.2\$ and for a maximum price of 4\$ per pound. The minimum quantity it is willing to sell is 1 pound and the maximum is 100. Then, agent *Seller* negotiation area is a triangle delimited by the lines connecting the points $(1.2, 1)$, $(4, 1)$, and $(1.2, 100)$; after each failed interaction, it will vary the offer by a δ parameter fixed to 0.03. Agent *Buyer* is interested to buy the product for a minimum price of 1\$ and a maximum of 3\$ per pound, while the quantity of the product it is willing to buy goes for a minimum of 1 pound to a maximum of 90. Then, its negotiation area is a rectangle represented by the constraints $X \geq 30, 90 \geq X, Y \geq 1, 3 \geq Y$; in want of an agreement, it will vary the proposal by a parameter $\delta = 0.02$. The intersection of the feasibility regions is not empty, and thus an agreement will be finally reached.

At this experiment 500 instances of such case are introduced, and the results obtained are as follows:

- (i) the average number of interactions necessary to conclude the process is 6.5;
- (ii) the minimum number of interactions observed during the experiments is 1, with a frequency of 56% (i.e., 56 negotiation processes out of 100 ended in just 1 step with an agreement);
- (iii) the maximum number of interactions observed is 15, with a frequency of 3%.

We have then tried to understand which parameters of the process can influence, and if so in which way, the number of

interactions needed to conclude the process. The significant parameters are at least (i) the size of the intersection area, (ii) the distance between the first proposals of the two agents, (iii) the size of the negotiation area, and (iv) the choice of the circumference radius.

In all experiments, there are considered 500 instances per each subcase

Case i: The Size of the Intersection Area. One may notice that the number of iterations increases with the decrease of the size of the intersection area: in fact, the average number of iterations tends to be inversely proportional with respect to the size of the parameter.

Case ii: Distance between First Proposals. The performance is likely to be influenced by the distance between the first proposals of the two agents. To verify this hypothesis it is assumed that the agent's A negotiation area is represented by the set $C_A = \{X \geq 3, X \leq 33, Y \geq 370, Y \leq 740\}$ (and therefore the negotiation area of agent A is 30 by 370), and the δ parameter has been set to 12. Agent B instead has an area of 15 by 130, and the δ parameter is set to 10.

First, proposals are considered to be random points of the negotiation areas, and they are forced to be at the maximum possible distance. Then, the size of the intersection area is varied in order to make a point on the algorithms behavior.

First Proposals as Random Points. The initial intersection area is $I_x * I_y = 1 * 1$.

First Proposals at Maximum-Distance Points. The situation is similar to the previous one (i.e., negotiation area, parameter δ , and initial intersection area are the same) with one variation: this time agents are forced to choose, as first offer, the point which is more distant from the intersection area. In this way, the first two proposals are considered to be at the maximum possible distance.

By comparing the average, the maximum, and the minimum numbers of interaction where first proposals are random points of the negotiation areas or instead where first proposals are considered to be more distant, it may be noticed that the distance between the first proposals actually affects the algorithm performance.

In the former case, where the first proposals are random points, the average number of iterations roughly decreases with the increase of the size of the intersection area, as the random factor prevents a more decided correlation. However, the statistics table shows that most instances succeed within a low-medium number of iterations.

In the latter case, where there is the maximum possible distance between the first proposals, the average number of iterations clearly decreases with the increase of the size of the intersection area. However, most instances succeed within a medium-high number of step, except when the intersection area becomes really wide.

Case iii: The Size of the Negotiation Area. With this third experiment we assessed the role that the variation of the negotiation areas can play. To this aim, we have considered two

agents represented by their (polyhedral) feasibility regions; during this test, regions are going to vary. The δ parameter has been set to 5 (for both agents), and the intersection area has been set to 5 by 5. That is, the feasibility regions are large while the intersection area is comparatively small and δ is rather small.

The average number of steps increases with the size of the negotiation areas, which however is what one may reasonably expect. So, as a rule of thumb we propose to cope with large areas by increasing the δ margin so as to proceed towards the opponent with *longer* steps.

Case iv: The Choice of δ . Noticeably, the increase of the number of iterations can be represented as a curve, where the number of iterations will be high in cases where δ takes extreme values (i.e., either very small or very high values compared to the size of the feasibility regions). The likely interpretation is that with very small δ 's each agent has to go a long way to approach the opposer's proposal; thus many *steps* will be needed. If instead δ is too big, say, bigger than the intersection area, then some steps can *jump* the intersection and force to randomly walk around the intersection. For the given feasibility regions which are rectangles, the best value for δ appears to be a bit less than the smaller edge. In fact, even starting from a random point there is a good chance to meet the intersection quickly.

4.1. Comparison with the Original RMP Approach. The last experiment mentioned previously is useful for a comparison between the extended algorithm ERMP and the original RMP. It should be noticed that the extended algorithm is able to cope with the first problematic situation discussed in Section 2, and, in general, it produces a more accurate solution. In fact, as both agents are allowed to adopt ERMP, they choose subsequent proposals by considering the opponent's previous ones. Then, they proceed "one towards the other" thus possibly reaching an agreement even when they have no vertices to offer. Moreover, if starting from preferred vertices the potential agreement will more probably fulfill their private utilities.

With respect to the basic approach, the performance of the extended algorithm is worse in case of negotiation spaces with a limited number of vertices, but works better, in average, in the opposite case (high number of vertices). This feature addresses one of the main challenges in developing effective negotiation protocols, that is, scalability; often in fact a protocol (including RMP) can produce excessively high failure rates, when there are many possible offers, due to computational intractability.

5. Related Work

The interesting survey about automated negotiation by Beam and Segev [24] mentions a famous negotiation process that occurred in the past: in 1899, Leo Baekeland sold the rights of his invention, Velox photographic printing paper, to Eastman Kodak for a \$1 million. Since then, negotiation procedures and strategies have certainly evolved, and since recently the exchange of proposals among humans is being at least

partly replaced by automated processes involving networked computers.

The first negotiation support systems (NSSs) [25] date back to 1980 and were aimed at assisting the users to take better decisions and to negotiate more productively. In the same period, important aspects of the negotiation have been explored. Raiffa [26] investigates decision-theoretic issues, Fisher and Ury [27] concentrate their work on win-win strategies, and Mastenbroek [28] considers the emotional aspects of negotiation processes.

While NSS was aimed at supporting the users during their negotiation processes, in the last years the first fully automated systems were proposed: in these systems, negotiation involves intelligent agents, that is, entities capable of a certain autonomy. Sycara and Zeng [29] in 1996 outline a metaframework for coordinating and structuring a collection of intelligent agents. In the same year, Chavez and Maes [30] created Kasbah, a marketplace for negotiating the purchase and sale of goods using intelligent software agents. What makes Kasbah fundamentally interesting is the autonomy of the agents and the interaction and competition between them in the marketplace.

After Kasbah, a variety of architectures are proposed, among them are [31–35]. These architectures can be seen as divided into *argumentation-based*, *heuristic-based*, *game-theoretic*, and *proposal-based*.

Game-theoretic techniques are based on strategies which have been extensively studied in game theory. The heuristics are mainly based on empirical testing and evaluation. In argumentation-based approaches, the negotiating parties can exchange any kind of feedback rather than just proposals and counterproposals. Among them we mention [36–39]. In [39], negotiation is defined as a single-issue process. The speed of negotiation is tackled in [38], but in this case variables are Boolean while in our approach they are real numbers.

We may notice that ERMP can be adopted either as a stand-alone strategy of negotiation or as a constraint-based technique to be used in the context of more general architectures like, for instance, the one in [37]. There is however a basic difference between our approach and other related research such as the one in [37]: they support both *one-to-one* and *one-to-many* negotiation, without however considering complexity issues. Instead, our approach is mainly aimed at obtaining a reasonable performance in terms of number of interactions steps.

Faratin et al. in [36] have presented a strategy, called *tradeoff*. This work deals with a multi-issue negotiation process and considers efficiency in terms of the total time used by the algorithm to conclude the process. The main difference of this approach with respect to the one presented here is that in [36] proposals are chosen according to their “similarity” with the last counteroffer. To do so, they use an algorithm (invoked repeatedly at each step) that selects the most similar tradeoff (to make) by moving down at various isocurve levels. Thus, they try to speed-up the process of negotiation by minimizing the number of interactions.

A recent research work that proposes intelligent agents for negotiation in the fashion business field is that of [10]. It considers an agent-based solution to a make-to-order

fashion supply chain negotiation scenario, mostly related to two issues: due date and pricing. They make the realistic assumption that downstream manufacturer and the upstream supplier are cooperative on due-date and competitive on price. In their view, the advantage of adopting agents is on the one hand saving time and on the other hand avoiding human negotiators’ subjective biases. They propose a two-phase negotiation strategy, respectively, on competitive pricing and cooperative due-date bargaining, with the aim of reaching Pareto efficiency of the overall process. In particular, their approach combines linear programming modeling with intelligent algorithms. The intelligent part of the method achieves Pareto efficiency by compensating the possible loss of one party on one issue by concessions of the other one, where a sensitivity analysis copes with the impact of probable variation of parameters and inputs.

6. Application to Fashion and Textile Supply Chains

This section briefly discusses our RM framework in the light of the specific needs of the fashion and textile (FT) supply chain management. As discussed, for example, in [11, 40, 41], and others (see especially Sycara’s survey [13] and the references therein), negotiation has been studied in both scientific and social sciences communities, by investigating different aspects of negotiation with different goals: social scientists try to understand the factors and reasoning processes that underlie human negotiation behavior. Computer scientists try to formulate mathematical models that capture elements of negotiation. Mathematical models can either be analytic (as in economics and operational research) or computational. The aim of the analytic models is to provide managerial guidance to optimize negotiation activity. The computational models (like, e.g., those discussed in this paper) aim to achieve computational tractability through approximation algorithms and heuristics. Computational models are nowadays most often implemented by means of autonomous agents and MAS solutions.

As argued in [13] computational models, though relying on approximate algorithms and heuristics, have still the flexibility to include cognitive considerations and features of human reasoning. Presently, and in perspective, such models could be used for decision support of human decision makers, and in the long run, such models (implemented by agents) can even substitute human mediators or negotiators.

The specific issues of this supply chain scenario are both promising and potentially problematic, we believe, for all agent-based approaches to management. First of all, the online nature of the problem, with its almost-real-time reaction requirements, is against analytic models which, by being high-level, pay a price (if implemented) in terms of raw performance. On the other hand, the changeable nature of the supply chain and the fact that bidders may enter or leave a call for tender at any time call exactly for computational models where the compositionality property, that is, the formalization is fully parametric with respect to the bids, is guaranteed.

One feature that is rather specific to the fashion and textile supply chain is the fact that negotiation is in general not needed in all circumstances and on all issues. For example, the buyer (e.g., a fashion retailer) may especially predefine the terms, products specification, fabric, production quantity, and so forth and ask for the quotation from a panel of selected sellers (i.e., the manufacturers). Usually, the one who quotes the lowest price upfront will be the final sellers.

However, following the analysis of Bruce and Daly [42, 43], there are still possible scenarios for negotiation in the fashion and textile supply chain, among which, for instance, the following:

- (i) the retailer requires a specific technology to produce an innovative fashionable product, and the number of this kind of sellers is limited in the market;
- (ii) the availability of raw material (e.g., cotton) and price fluctuate significantly.

As emphasized in [10] (see also the references therein), in a supply chain context, negotiation is a complex issue because not only a lot of factors have to be considered, but also incomplete information and changing circumstances should be taken into account. Thus, in perspective intelligent negotiation algorithms will have to cope with the complexity and uncertainty in real-world negotiation and with its impact on the negotiators' motivations. In general, in fact, negotiating agents have motivations. Social sciences have adopted the distinction between selfish and prosocial motivations. Selfish motivation is characterized by competitive and individualistic goals, while prosocial motivation is characterized by cooperative and altruistic goals. In many realistic settings, a negotiator has mixed motives rather than behaving purely selfishly or purely prosocially. These issues can be identified in the FT supply chain. In fact, seeking optimal decisions in terms of production scheduling, product quality control, and final shipping and distribution can be seen as prosocial motivations, at least within a company or a coordinated supply chain.

Market requirements on fast fashion and short lead time call instead for efficient computational models such as the ones discussed in this paper. In these models, an agents' motivation is usually to maximize its own utility in terms of utility function or preferences structure while staying within almost-real-time time bounds. We believe that, at present, efficient computational models (such as ERMP and the one of [10]) can greatly ease the work of human managers (both sellers and buyers) who, to respect the time-to-market aspect, have to negotiate every day with multiple counterparts on multiple issues. There has been in fact consistent evidence that using intelligent agents to negotiate instead of or with humans achieves better outcomes than negotiation between two human beings (see, e.g., Lin et al. [9] and the references therein).

In perspective, however, an integration with analytic methods might better contribute to a full (or almost full) automation of a whole supply chain framework. A promising direction is that proposed by Sycara and Dai [13] to use persuasive argumentation to influence the other sides' belief

structure. The purpose is to influence the other parties' utility function, including goals, importance attached to different goals, and relations between goals. Ultimately, argumentation can either change the opponents' utility value of one objective or change the relative importance he assigns to that objective.

7. Conclusions and Future Work

The ERMP model proposed in this paper is based upon a heuristic algorithm that considers not only the vertices as possible offers but also internal points of the feasibility regions. We have shown that the extended approach can be usefully combined with the original one, thus obtaining a local search variant that is able to cope with, for example, preferences, while still aiming at minimizing the number of steps.

By comparing the proposed algorithm with the Cadoli's original RMP [5], we showed how our work overcomes some problems of the basic approach. Unfortunately, in our case the number of interactions is no longer proportional to the number of vertices. However, experiments show that the proposed algorithm has a reasonable performance in the average case. Moreover, in some cases—notably when many vertices are involved in the negotiation—it can be even more efficient than the original one. Though convergence is not in general guaranteed, we have defined sufficient conditions for convergence with no additional computational cost. As discussed before, one can either impose an upper bound to the allowed number of steps or consider only a finite number of points of the feasibility region. In the experiments we have adopted an upper bound, which however has never been reached in experiments where there is a possible agreement.

The additional complexity, according to the experiments, appears to be a reasonable price to pay for the extra features and for the possibility of other extensions. In fact, the approach can be further extended, for example, by adding new protocols and objective and utility functions. We in fact envisage an integration with [8]. We also intend to consider peer-to-peer negotiation where the parts involved in the process use more than two issues, involving in such way a multidimensional space. We have been studying the possibility of considering as negotiation spaces not only convex areas but also nonconvex ones. This result can be obtained by converting a nonconvex region into a convex one and by excluding all points that are not part of the original negotiation area [44]. The proposed approach has been fully implemented in the agent-oriented programming language DALI.

Applications of such an efficient and effective computational model of negotiation can be found in many fields where there is a large number of negotiation rounds, almost impossible to be optimally coped with by humans. These fields include contexts where, as it is the case with the fashion supply chain negotiation scenario, the time-to-market aspect is crucial.

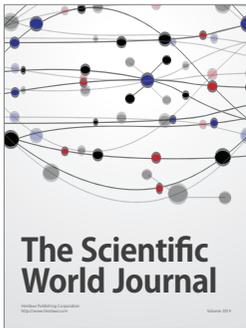
Acknowledgment

This work started as a cooperation with Marco Cadoli. Unfortunately, Marco left us in November 2006 after a long illness. This paper is dedicated to him.

References

- [1] G. Governatori, A. H. ter Hofstede, and P. Oaks, "Defeasible logic for automated negotiation," in *Proceedings of the COLLECTeR*, Deakin University, 2000.
- [2] D. N. Walton and E. C. W. Krabbe, *Commitment in Dialogue*, SUNY Series in Logic and Language, State University of New York Press, Albany, NY, 1995.
- [3] N. R. Jennings, S. Parsons, C. Sierra, and P. Faratin, "Automated negotiation," in *Proceedings of 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Systems*, 2000, <http://www.scientificcommons.org/23363630>.
- [4] S. Costantini, A. Tocchio, and P. Tsintza, "A heuristic approach for p2p negotiation," in *Proceedings of 8th Workshop on Computational Logic in Multi-Agent Systems (CLIMA '07)*, 2007.
- [5] M. Cadoli, "Proposal-based negotiation in convex regions," in *Proceedings of the Cooperative Information Agents VII (CIA '03)*, 2003, <http://www.springerlink.com/content/a0979twuhv099rw4>.
- [6] R. G. Smith, "The contract net protocol: high-level communication and control in a distributed problem solver," *IEEE Transactions on Computers*, vol. 29, no. 12, pp. 1104–1113, 1980.
- [7] H. Ehtamo, E. Kettunen, and R. P. Hämäläinen, "Searching for joint gains in multi-party negotiations," *European Journal of Operational Research*, vol. 130, no. 1, pp. 54–69, 2001.
- [8] T. Mancini, "Negotiation exploiting reasoning by projections," in *Proceedings of the 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS '09)*, Y. Demazeau, J. Pavón, J. M. Corchado, and J. Bajo, Eds., vol. 55 of *Advances in Intelligent and Soft Computing*, Springer, 2009.
- [9] R. Lin, S. Kraus, J. Wilkenfeld, and J. Barry, "Negotiating with bounded rational agents in environments with incomplete information using an automated agent," *Artificial Intelligence*, vol. 172, no. 6–7, pp. 823–851, 2008.
- [10] A. Pan and T.-M. Choi, "An agent-based negotiation model on price and delivery date in a fashion supply chain," *Annals of Operations Research*, 2012.
- [11] G. Lai, K. Sycara, and C. Li, "A decentralized model for multi-attribute negotiations with incomplete information and general utility functions," *Multiagent and Grid Systems*, vol. 4, no. 1, pp. 45–65, 2008.
- [12] P. Heiskanen, H. Ehtamo, and R. P. Hämäläinen, "Constraint proposal method for computing Pareto solutions in multi-party negotiations," *European Journal of Operational Research*, vol. 133, no. 1, pp. 44–61, 2001.
- [13] K. Sycara and T. Dai, "Agent reasoning in negotiation," in *Handbook of Group Decision and Negotiation*, vol. 4 of *Advances in Group Decision and Negotiation*, Springer, 2010.
- [14] K. Somefun, E. Gerding, S. Bohte, and H. La, "Automated negotiation and bundling of information goods," in *Proceedings of the 5th Workshop on AgentMediated Electronic Commerce (AMEC '03)*, Melbourne, Australia, July 2003.
- [15] K. Fujita, T. Ito, and M. Klein, "The effect of grouping issues in multiple interdependent issues negotiation based on cone-constraints," in *New Trends in Agent-Based Complex Automated Negotiations*, T. Ito, M. Zhang, V. Robu, S. Fatima, and T. Matsuo, Eds., vol. 383 of *Studies in Computational Intelligence*, Springer, 2012.
- [16] E. Aarts and J. K. Lenstra Segev, *Local Search in Combinatorial Optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, Chichester, UK, 1997.
- [17] S. Costantini and A. Tocchio, "A logic programming language for multi-agent systems," in *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA '12)*, vol. 2424 of *Logics in Artificial Intelligence*, Springer, Berlin, Germany, 2002.
- [18] S. Costantini and A. Tocchio, "The dali logic programming agent-oriented language," in *Proceedings of the 9th European Conference (JELIA '04)*, vol. 2424 of *Logics in Artificial Intelligence*, Springer, Berlin, Germany, 3229.
- [19] S. Costantini and A. Tocchio, "About declarative semantics of logic-based agent languages," in *Proceedings of the Declarative Agent Languages and Technologies (DALT '06)*, vol. 3229 of *Logics in Artificial Intelligence*, Springer, Berlin, Germany, 2006.
- [20] S. Costantini, A. Tocchio, and A. Verticchio, "A game-theoretic operational semantics for the dali communication architecture," in *Proceedings of the Workshop on Objects and Agents (WOA '04)*, Turin, Italy, December 2004.
- [21] A. Tocchio, *Multi-agent systems in computational logic [Ph.D. thesis]*, Dipartimento di Informatica, Università degli Studi di L'Aquila, 2005.
- [22] A. Kowalski, *How to be Artificially Intelligent—the Logical Way*, 2004, <http://www-lp.doc.ic.ac.uk/UserPages/staff/rak/rak.html>.
- [23] J. W. Lloyd, *Foundations of logic Programming*, Symbolic Computation. Artificial Intelligence, Springer, Berlin, Germany, 2nd edition, 1987.
- [24] C. Beam and A. Segev, "Automated negotiations: a survey of the state of the art," *Wirtschaftsinformatik*, vol. 39, no. 3, pp. 263–268, 1997.
- [25] L. H. Lim and I. Benbasat, "A theoretical perspective of negotiation support systems," *Journal of Management Information Systems*, vol. 9, no. 3, pp. 27–44, 1992.
- [26] H. Raiffa, *The Art and Science of Negotiation*, Harvard University Press, Cambridge, Mass, USA, 1982.
- [27] R. Fisher and W. Ury, *Getting to Yes: Negotiating Agreement without Giving In*, Houghton Mifflin, Boston, Mass, USA, 1981.
- [28] W. Mastenbroek, *Negotiate*, Blackwell, Oxford, UK, 1989.
- [29] K. Sycara and D. Zeng, "Coordination of multiple intelligent software agents," *International Journal of Cooperative Information Systems*, vol. 5, no. 2–3, pp. 181–211, 1996.
- [30] A. Chavez and P. Maes, "Kasbah: an agent marketplace for buying and selling goods," in *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM '96)*, pp. 75–90, Practical Application Company, London, UK, 1996.
- [31] J. Collins, B. Youngdahl, S. Jamison, B. Mobasher, and M. Gini, "Market architecture for multi-agent contracting," in *Proceedings of the 1998 2nd International Conference on Autonomous Agents*, pp. 285–292, May 1998.
- [32] J. Eriksson and N. Finne, *Marketspace: an open agentbased market infrastructure [M.S. thesis]*, Computing Science Department, Uppsala University, 1997.
- [33] J. A. Rodriguez, P. Noriega, C. Sierra, and J. Padget, "Fm96.5 a java-based electronic auction house," in *Proceedings of the 2nd International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM '97)*, T. P. A. Company, Ed., pp. 207–224, The Practical Application Company, London, UK, 1997.
- [34] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons, "A framework for argumentation-based negotiation," in *Proceedings of the 4th International Workshop on Agent Theories, Architectures, and Languages (ATAL '97)*, vol. 1365 of *Lecture Notes in Computer Science*, pp. 177–192, 1998.

- [35] M. Tsvetovatyy, M. Gini, B. Mobasher, and Z. Wieckowski, "Magma: an agent-based virtual market for electronic commerce," *Applied Artificial Intelligence*, vol. 11, no. 6, pp. 501–523, 1996.
- [36] P. Faratin, C. Sierra, and N. R. Jennings, "Using similarity criteria to make issue trade-offs in automated negotiations," *Artificial Intelligence*, vol. 142, no. 2, pp. 205–237, 2002.
- [37] I. Rahwan, R. Kowalczyk, and H. H. Pham, "Intelligent agents for automated one-to-many ecommerce negotiation," in *Proceedings of the 25th Australasian conference on Computer Science (ACSC '02)*, pp. 197–204, Australian Computer Society, Darlinghurst, Australia, 2002.
- [38] M. Wooldridge and S. Parsons, "Languages for negotiation," in *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI '00)*, pp. 393–400, 2000.
- [39] M. Yokoo, E. H. Durfee, T. Ishida, and K. Kuwabara, "The distributed constraint satisfaction problem: formalization and algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 10, no. 5, pp. 673–685, 1998.
- [40] L.-M. Hofmann, N. Chakraborty, and K. P. Sycara, "The evolution of cooperation in selfinterested agent societies: a critical study," in *Proceedings of the of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '11)*, pp. 685–692, 2011.
- [41] B. D. Ziebart, M. Dudík, G. J. Gordon, K. P. Sycara, W. L. Adair, and J. Brett, "Identifying culture and leveraging cultural differences for negotiation agents," in *Proceedings of the 45th Hawaii International International Conference on Systems Science (HICSS '12)*, pp. 618–627, IEEE Computer Society, 2012.
- [42] M. Bruce and L. Daly, "Buyer behaviour for fast fashion," *Journal of Fashion Marketing and Management*, vol. 10, no. 3, pp. 329–344, 2006.
- [43] M. Bruce and L. Daly, *Challenges of Fashion Buying and Merchandising*, Fashion Marketing, 2007.
- [44] A. Siegel, "A historical review of the isoperimetric theorem in 2-d, and its place in elementary plane geometry," 2003, <http://www.cs.nyu.edu/faculty/siegel/SCIAM.pdf>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

