Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT

Artificial Intelligence 148 (2003) 53–102

Artificial Intelligence

www.elsevier.com/locate/artint

A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments

Xudong Luo [a,*], Nicholas R. Jennings [a], Nigel Shadbolt [a], Ho-fung Leung [b], Jimmy Ho-man Lee [b]

[a] *Department of Electronics and Computer Science, The University of Southampton, Highfield, Southampton SO17 1BJ, UK*

[b] *Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, People's Republic of China*

Received 2 July 2001; received in revised form 11 July 2002

Abstract

This paper develops a fuzzy constraint based model for bilateral multi-issue negotiation in trading environments. In particular, we are concerned with the principled negotiation approach in which agents seek to strike a fair deal for both parties, but which, nevertheless, maximises their own payoff. Thus, there are elements of both competition and cooperation in the negotiation (hence semi-competitive environments). One of the key intuitions of the approach is that there is often more than one option that can satisfy the interests of both parties. So, if the opponent cannot accept an offer then the proponent should endeavour to find an alternative that is equally acceptable to it, but more acceptable to the opponent. That is, the agent should make a trade-off. Only if such a trade-off is not possible should the agent make a concession. Against this background, our model ensures the agents reach a deal that is fair (Pareto-optimal) for both parties if such a solution exists. Moreover, this is achieved by minimising the amount of private information that is revealed. The model uses prioritised fuzzy constraints to represent trade-offs between the different possible values of the negotiation issues and to indicate how concessions should be made when they are necessary. Also by using constraints to express negotiation proposals, the model can cover the negotiation space more efficiently since each exchange covers a region rather than a single point (which is what most existing models deal with). In addition, by incorporating the notion of a reward into our negotiation model, the agents can sometimes reach agreements that would not otherwise be possible.

* Corresponding author.
E-mail address: xl@ecs.soton.ac.uk (X. Luo).

0004-3702/$ – see front matter
doi:10.1016/S0004-3702(03)00041-9

Keywords: Software agents; Automated negotiation; Fuzzy constraint satisfaction; E-business

1. Introduction

Negotiation is a process by which a group of entities try and come to a mutually acceptable agreement on some matter [44]. Because of its ubiquity in everyday encounters, it is a subject that has been extensively discussed in the game-theoretic, economic, and management science literatures [45,46,69]. Recently, however, there has been a surge of interest in automated negotiation systems that are populated with artificial agents [20]. This is due to both a technology push and an application pull [50]. The technology push is mainly from a growing standardised communication infrastructure (e.g., the Semantic Web and the Grid) which allows distributed and heterogeneous entities to interact flexibly. The application pull is from domains (e.g., supply chain management, telecommunication network management, virtual organisations and electronic trading systems) that require self-interested software entities, representing different stakeholders, to interact in a flexible manner. In these applications, conflicts often arise because the agents represent distinct stakeholders with different perspectives and different preferences. Allied to this, is the fact that the agents act autonomously (i.e., they decide for themselves what actions they should take, at what time, and under what terms and conditions [60]). In such circumstances, the interaction between the agents can only proceed by the process of making proposals and/or trading offers, with the aim of finding a mutually acceptable agreement. In short, by *negotiation*.

The process of automating negotiations also opens up a number of new possibilities. In contrast to its manual counterpart, the potential advantages of automated negotiation are as follows:

(1) Manual negotiation of contracts is time consuming and hence expensive. Thus, it tends to be carried out relatively infrequently. This inertia means that institutions tend to stay locked into contracts that may not be in either parties best interest. In contrast, by automating the process, negotiations can take place much more frequently, between many more partners, for much smaller value goods. This has the effect of making commerce much more frictionless and responsive to the prevailing circumstances which should make it more efficient.
(2) Manual negotiation is often considered either too embarrassing or frustrating [37] for ordinary consumers (even if it is in their best interest to do so). Automation removes these human sensibilities and can lead to more satisfactory outcomes. Moreover, complicated negotiation problems (perhaps involving multiple, inter-related goods) are often too difficult for many consumers to handle manually. In this case, automated negotiation systems can help ordinary users perform like experts in complicated negotiations (see [5] for preliminary evidence in this direction).
(3) Automated negotiations do not require the participants to be colocated in space or time. This means that the number of entities with which an agent can negotiate is increased.

This, in turn, should improve the likely outcome for both buyers and sellers (see [1] for a full discussion).

According to the cardinality and nature of the interaction, automated negotiation models can be classified into three main categories [20]. The first consists of many-to-one or many-to-many models in which multiple agents negotiate with either a single or many other agents. This category is predominantly handled using various auction-based models [17,50,58] and these models are widely used in the field of on-line retail (e.g., eBay (http://www.ebay.com) and *e*Mediator [51]. The second category consists of one-to-one models in which a pair of agents negotiate directly with one another [2,9,27,39,43]. These models typically use a range of heuristic methods to cope with the uncertainties that are endemic in encounters of this type (see Section 6 for more details of this type of model). The third category consists of argumentation/persuasion-based models [28,29,42] in which agents use various types of argument, such as *threats*, *rewards* and *appeals*, to persuade their opponent to accept a deal they would not previously have countenanced. For each of the three categories, the negotiation domain could be a single-issue one (e.g., price) or a multiple-issue one (e.g., price, quality, model, volume, delivery date, expiry date, after-sale service, warranty and return policy).

In this paper, we concentrate on the one-to-one case and develop an automated model for multi-issue negotiations. In particular, we are interested in trading agents in retail markets and so we cast our model in terms of buyer and seller agents. In such markets, it is unnecessary for the sellers to be hostile to buyers [16]. It is more important to make the customer as satisfied as possible so that long-term relationships can be established. Such relationships should then ensure long-term profitability. The main business negotiation theory suitable for this kind of environment is the *principled negotiation approach* [3, 11,19,45]. In this method, agents strive to reach a fair and reasonable agreement for both parties, but which, nevertheless, maximises their own payoff. Thus, it can be seen that principled negotiation has elements of both competition (the agents simply maximise their individual payoff without regard to the outcome for their opponent) and cooperation (the agents want to maximise the outcome for both participants). Thus we reterm the environment *semi-competitive*.

To base our model on the principled negotiation approach, our design should fulfill a number of requirements. Firstly, the solutions should be fair for both sides. Secondly, both sides should endeavour to maximise their own payoff during the course of a negotiation. Thus, if the opponent cannot accept an offer then the proponent should endeavour to find an alternative that has the same value to it (i.e., the agent should trade-off between the various negotiation issues [9]). In other words, an agent should avoid making a concession, if possible, since this lowers its payoff from the deal. Moreover, when an agent does have to make a concession it should make the smallest one possible. Thirdly, it is important that the agents minimise the amount of information they reveal about their preferences since any such revelation can weaken their bargaining position [45,46].[1] Finally, in many real

[1] Another reason why the agents need to minimise such information revelation is that humans are basically unwilling to disclose private information during a negotiation [18]. Thus, if we want negotiation agents to be generally acceptable, they must follow the same broad tenet.

retail markets, sellers often use gifts to raise a products' acceptability for the customers [4]. This gift may make a previously unacceptable offer acceptable.

Our model employs the notion of prioritised fuzzy constraints [6–8,32,34] at its core (in particular, to determine which offer should be sent, whether an offer is acceptable, and which counter offer should be made). The notion of Prioritised Fuzzy Constraint Satisfaction Problems (PFCSPs) was chosen as the basis of our negotiation model for the following reasons:

(1) In many cases, buyers do not know the precise details of the products they want to buy, and so their requirements are often expressed by constraints over multiple issues (or "attributes"). For example, consider the case of an international student who just arrives in the UK for the first time and who has to rent some accommodation. Since he is totally new to the country, he cannot tell a real estate agent exactly what he wants, but he can naturally express his requirements as some constraints (e.g., the accommodation should be within walking distance of the university, the rent should not be too high, and it would be better if there is an Internet connection).
(2) When buyers and sellers negotiate, it is rarely the case that an offer is completely acceptable or completely inconsistent with their respective constraints. Rather, an offer usually satisfies the buyer's constraints more or less. For example, an offer from the real estate agent, 25 minutes walk and 300 pounds per month, can partially satisfy the student's constraints because the distance is a little too far but is still just about acceptable, and considering his limited scholarship the rent is not beyond his budget but a lower one would be more acceptable. The PFCSP framework is ideally suited for capturing constraints of this kind because fuzzy constraints can be partially satisfied or violated. In fact, the student's constraints *are* fuzzy constraints.
(3) For a single attribute of the desired product, a buyer might prefer certain values over others (e.g., for accommodation type, the student prefers "single room in a flat" over "shared room in a house"). Such a preference can be expressed as a fuzzy constraint over a single attribute, and the preference level at a certain value of the attribute is the constraint's satisfaction degree for that value. Similarly, for multiple product attributes, a buyer might prefer certain combinations of values over others (e.g., for rental and period, the student prefers "cheap and short contracted period" over "expensive and long contracted period"). Such a preference can be expressed as a fuzzy constraint over multiple attributes, and the preference level at a certain combination value of these attributes is the constraint's satisfaction degree for the combination value.
(4) One of the fundamental things in negotiation is to represent trade-offs (balances) between the different possible values for attributes. A buyer's preferences on trade-offs between different attributes of the desired product can easily be modelled by fuzzy constraints. For instance, consider the trade-off involved in an agent deciding whether it prefers to get exactly the desired value of an attribute that is very important or several sets of less good values for attributes that are less important to it. Such a trade-off can be modelled by a fuzzy constraint. For example, in an accommodation renting scenario, suppose that the distance is the most important thing, then rental rate and finally flexible rental period. Now suppose the ideal distance is about 15 minutes to the university, rental rate 300 pounds or so is acceptable and at least 6 months contract

is acceptable. However, if the distance becomes 50 minutes but the rental rate becomes 150 pounds and no restriction is placed on rental period, such an accommodation is still acceptable. For these combinations of attributes' values, we simply assign the same satisfaction degree (the fuzzy constraint is over attributes distance, rental rate and rental period). If there are some other trade-offs, the student's preference for them can be modelled by assigning them different satisfaction degrees (the bigger, the more preferable).

(5) A buyer's constraints are not always equally important. For example, the student may think that the constraint on rent is more important than the one on distance. In order to deal with different levels of importance of different fuzzy constraints, Dubois et al. [6–8] introduced the concept of priority into Fuzzy CSPs (FCSPs) to form PFCSPs.

This work advances the state of the art in the following main ways. Firstly, our model ensures the agents reach a fair deal (a Pareto optimal solution) for both parties if such a solution exists. Moreover, this is achieved by minimising the amount of private information that both sides have to reveal. Secondly, by using constraints to express offers, negotiation proposals can cover regions of the solution space rather than just single point solutions. This, in turn, means that more of the space can be explored in a given exchange and so means that the search for a mutually acceptable solution is more efficient. Thirdly, by incorporating the notion of a reward into our negotiation model, the agents can sometimes reach agreements that would not otherwise be possible.

The remainder of the paper is organised as follows. Section 2 presents the basic concepts and notations related to constraint satisfaction problems and fuzzy mathematics that will be used throughout the paper. Section 3 defines the negotiation model of the seller and the buyer agents. The model consists mainly of their domain knowledge, primitive actions, behaviour protocols and communication language. Section 4 explores the properties of our negotiation model. Section 5 illustrates the operation of our model with an accommodation renting scenario. Section 6 compares our model with related work. Finally, Section 7 summarises our main contributions and indicates avenues of further research.

2. Preliminaries

In this section, we recall the necessary concepts and notations related to constraint satisfaction problems [36,55], fuzzy constraint satisfaction problems [6], prioritised fuzzy constraint satisfaction problems [6–8,32,34], uninorm operators [62], fuzzy logic [67,68] and the cut-set technique [24] in fuzzy mathematics. These provide the underpinning of our negotiation model.

We start with the framework of constraint satisfaction problems (CSPs).

Definition 1. A *constraint satisfaction problem* (*CSP*) is a 3-tuple (X, D, C), where sets:

(1) $X = \{x_i \mid i = 1, \ldots, n\}$ is a finite set of *variables*.
(2) $D = \{d_i \mid i = 1, \ldots, n\}$ is the set of *domains*. Each domain d_i is a finite set containing the possible values for the corresponding variable x_i in X.

(3) $C = \{R_i \mid R_i \subseteq \prod_{x_j \in var(R_i)} d_j, i = 1, \ldots, m\}$ is a set of *constraints*. Here $var(R_i)$ denotes the set of variables of constraint R_i:

$$var(R_i) = \{x'_1, \ldots, x'_{k_{R_i}}\} \subseteq X. \tag{1}$$

Definition 2. A *label* of a variable x is an assignment of a value to the variable, denoted as v_x. A *compound label* $v_{X'}$ of all variables in set $X' = \{x'_1, \ldots, x'_m\} \subseteq X$ is a simultaneous assignment of values to all variables in set X':

$$v_{X'} = (v_{x'_1}, \ldots, v_{x'_m}). \tag{2}$$

Definition 3. In a CSP (X, D, C), the *characteristic function* of $R_i \in C$,

$$\mu_{R_i} : \left(\prod_{x_j \in var(R_i)} d_j \right) \to \{0, 1\},$$

is defined as:

$$\mu_{R_i}(v_{var(R_i)}) = \begin{cases} 1 & \text{if } v_{var(R_i)} \in R_i, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

A *solution* to a CSP (X, D, C) is a compound label $v_X = (v_{x_1}, \ldots, v_{x_n})$ of all variables in X such that:

$$\min\left\{\mu_{R_i}(v_{var(R_i)}) \mid R_i \in C\right\} = 1. \tag{4}$$

When returning the value 1, the characteristic function of a constraint R_i signifies the absolute satisfaction of the constraint R_i over a compound label $v_{var(R_i)}$. A return value of 0 means the complete violation of the constraint R_i over a compound label. In other words, in a standard CSP a constraint either admits a compound label or not (this is called a crisp constraint). There is no intermediary situation. However, this formulation is too rigid for dealing with problems in which the satisfaction level of a constraint is not a simple zero-one matter. Thus, the notion of fuzzy CSP [6] is introduced.

Definition 4. A *fuzzy constraint satisfaction problem* (*FCSP*) is a 3-tuple (X, D, C^f), where X and D are the same as those in Definition 1, and C^f is a set of *fuzzy constraints*:

$$C^f = \left\{ R_i^f \mid \mu_{R_i^f} : \left(\prod_{x_j \in var(R_i^f)} d_j \right) \to [0, 1],\ i = 1, \ldots, m \right\}, \tag{5}$$

where $var(R_i^f)$ denotes the set of variables of R_i^f.

By using the cut-set technique in fuzzy mathematics [24], a fuzzy constraint can induce a crisp constraint.

Definition 5. Given the *cut level* $\sigma \in [0, 1]$, the *induced crisp constraint* R^c of a fuzzy constraint R^f is defined as:

$$\mu_{R^c}(v_{var(R^f)}) = \begin{cases} 1 & \text{if } \mu_{R^f}(var(R^f)) \geqslant \sigma, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Intuitively, a cut level for a fuzzy constraint is a kind of threshold: if the satisfaction degree to which a compound label satisfies a fuzzy constraint is not less than the threshold, the label is regarded as satisfactory with respect to the constraint; otherwise, it is regarded as unsatisfactory.

In a standard FCSP, each constraint has no priority or, equivalently, all constraints have the same level of priority (importance). However, this is not always true in practice. Thus, Dubois et al. [6–8] extended FCSPs by associating different constraints with different priorities, and hence introduced the concept of prioritised FCSPs (PFCSPs). In this work, we use a revised version [32,34] of their concept.[2]

Definition 6. A *prioritised fuzzy constraint satisfaction problem* (*PFCSP*) is a 4-tuple (X, D, C^f, ρ), where (X, D, C^f) is a FCSP (see Definition 4), called the *counterpart FCSP* of the PFCSP, and $\rho : C^f \to [0, \infty)$ is a *priority function*. Given a compound label v_X of all variables in X, its *overall satisfaction degree* is given by:

$$\alpha_\rho(v_X) = \min\left\{\left(\frac{\rho(R^f)}{\rho_{\max}}\right) \diamond \mu_{R^f}(v_{var(R^f)}) \mid R^f \in C^f\right\}, \tag{7}$$

where

$$\rho_{\max} = \max\{\rho(R^f) \mid R^f \in C^f\}$$

(hereafter unless otherwise specified, the symbol $\rho_{\max}$ always takes this meaning), and operator $\diamond : [0, 1] \times [0, 1] \to [0, 1]$, called a *priority operator*, satisfies:

(1) $\forall a_1, a_2, a_2' \in [0, 1], a_2 \leqslant a_2' \Rightarrow a_1 \diamond a_2 \leqslant a_1 \diamond a_2'$,
(2) $\forall a_1, a_1', a_2 \in [0, 1], a_1 \leqslant a_1' \Rightarrow a_1 \diamond a_2 \geqslant a_1' \diamond a_2$,
(3) $\forall a \in [0, 1], 1 \diamond a = a$, and
(4) $\forall a \in [0, 1], 0 \diamond a = 1$.

A *solution* to a PFCSP (X, D, C^f, ρ) is a compound label v_X of all variables in X such that:

$$\alpha_\rho(v_X) \geqslant \tau, \tag{8}$$

where $\tau \in [0, 1]$ is a predetermined value, called the *solution threshold* of the PFCSP.

Intuitively, the solution threshold τ means that if the overall satisfaction degree of a compound label is not less than the threshold, the label is acceptable as a solution; otherwise, it is not.

In addition to the above concepts related to constraints, two further concepts from fuzzy mathematics are also needed for our model. The first one is that of uninorm operators

[2] Our extension has a number of special properties that we need in our negotiation system. In particular, whether priorities of constraints are normalised or not, the constraint with the highest priority can act like a hard constraint: if a compound label completely violates it then the label cannot be accepted as a solution. Secondly, when all constraints in a PFCSP have the same priority (importance), the PFCSP degenerates to a FCSP. Finally, in dynamic situations, users can add or remove constraints without re-assigning priorities to all the existing constraints in the constraint set. None of these properties hold in the original model.

[61,62]. We use this kind of operator to help calculate the buyer's acceptability for a product that consists of multiple attributes. Intuitively, acceptability depends mainly on the overall satisfaction degree of all the buyer's requirement constraints. As a secondary factor, however, the product's level of acceptability may be increased by the seller offering an additional reward to accompany the purchase of the product in question. Accordingly, when a seller uses a reward to try and raise the buyer's acceptability, if the acceptability level is originally less than the acceptability threshold, the reward can raise the acceptability to some extent but probably not to its maximal value. Similarly, if the acceptability level is originally greater than the acceptability threshold, the reward can raise the acceptability to some extent but cannot raise it beyond the maximum value. To capture these intuitions, uninorm operators are employed because they have a number of relevant properties (especially property (3) in Lemma 2 in the following). We will formally prove this point (see Section 4) after formally defining the buyer's acceptability for a product (see Section 3.2).

Definition 7. A binary operator $\oplus : [0,1] \times [0,1] \to [0,1]$ is called a *uninorm operator* if it is increasing, associative and commutative and there exists $\tau \in [0,1]$ such that:

$$\forall a \in [0,1], \quad a \oplus \tau = a. \tag{9}$$

Here τ is said to be the *unit element* of a uninorm.

The following lemma, related to uninorm operators, is useful later on, and so is listed here.

Lemma 1. *A uninorm operator $\oplus$ with unit element τ has the following properties*:

(1) $\forall a_1, a_2 \in (\tau, 1], a_1 \oplus a_2 \geqslant \max\{a_1, a_2\}$;
(2) $\forall a_1, a_2 \in [0, \tau), a_1 \oplus a_2 \leqslant \min\{a_1, a_2\}$;
(3) $\forall a_1 \in [0, \tau), a_2 \in (\tau, 1], \min\{a_1, a_2\} \leqslant a_1 \oplus a_2 \leqslant \max\{a_1, a_2\}$;
(4) $\forall a \in [0, 1], 0 \oplus a = 0$.

Actually, we can regard the unit element of a uninorm operator as a threshold: if an evaluation is greater than the threshold the evaluation is regarded as being *positive*; otherwise, it is regarded as being *negative*. Thus, in Lemma 1, property (1) reveals the intuition that when two evaluations are both positive they should enhance the effect of each other; property (2) that when two evaluations are both negative, they should weaken each other; and property (3) that when two evaluations are in conflict, we should get a compromise.

Given these general observations, the next step is to determine which specific uninorm operator we should adopt in our negotiation model. Here we chose the following[3] (from [23]):

[3] Our reason is as follows. For unit element $\tau \in (0,1)$, six other kinds of uninorm operator have been proposed: Yager and Rybalov [62] proposed R_* and R^*, and Li and Shi [30] proposed R_1, R_2, $\overline{R}$ and $\underline{R}$. Uninorm operators R_*, R_2 and $\underline{R}$ share the common characteristic: when $\min\{a_1, a_2\} \leqslant \tau \leqslant \max\{a_1, a_2\}$ the result of the operation on a_1 and a_2 is $\min\{a_1, a_2\}$. Uninorm operators R^*, R_1 and $\overline{R}$ share the common characteristic: when $\min\{a_1, a_2\} \leqslant \tau \leqslant \max\{a_1, a_2\}$, the result of the operation on a_1 and a_2 is $\max\{a_1, a_2\}$. In other words,

Lemma 2. *An operator* $\oplus_P : [0,1] \times [0,1] \to [0,1]$, *given by*:

$$a_1 \oplus_P a_2 = \frac{(1-\tau)a_1a_2}{(1-\tau)a_1a_2 + \tau(1-a_1)(1-a_2)}, \tag{10}$$

where $\tau \in (0,1)$, *is a uninorm operator with unit element* τ, *and*

$$\min\{a_1, a_2\} < \tau < \max\{a_1, a_2\} \quad \Rightarrow \quad \min\{a_1, a_2\} < a_1 \oplus_P a_2 < \max\{a_1, a_2\}. \tag{11}$$

The second concept from fuzzy mathematics that we exploit is fuzzy truth propositional logic [67,68]. Again this is useful in calculating the buyer's acceptability for a product. In particular, fuzzy truth propositions are employed to represent the facts in the buyer's profile model because these facts are often partially true (i.e., they have truth values between "completely true" and "completely false"). For instance, for a student who wants to rent accommodation, a telephone, new furniture and an economic heater might be valuable to degrees of 70%, 30% and 80%, respectively.

Definition 8. A *fuzzy truth proposition system* is 5-tuple $(F, t, \wedge, \vee, \neg)$, where

(1) $F = \{f_i \mid i = 1, \ldots, n\}$ is the set of *primitive propositions*,
(2) $t : F \to [0,1]$ is a *truth function*, which associates each proposition f with a *truth* $t(f) \in [0,1]$, and
(3) $\wedge$, $\vee$ and $\neg$ are logical operators.

A *composite proposition* c of primitive propositions $f_1, \ldots, f_n$, called a *Boolean expression* of $f_1, \ldots, f_n$, is constructed from $f_1, \ldots, f_n$ through the logical operators $\wedge$, $\vee$ and $\neg$. The *truth*, $t(c)$, of composite proposition c is calculated recursively by:

$$t(\neg c_0) = 1 - t(c_0), \tag{12}$$

$$t(c_1 \wedge c_2) = \min\bigl\{t(c_1), t(c_2)\bigr\}, \tag{13}$$

$$t(c_1 \vee c_2) = \max\bigl\{t(c_1), t(c_2)\bigr\}, \tag{14}$$

where c_0, c_1, c_2 are primitive propositions or composite propositions. The truth $t(c)$ of composite proposition c is also denoted as $c'(t(f_1), \ldots, t(f_n))$.

3. The negotiation model

In this section we present our conceptualisation of the negotiating agents. Firstly, we specify the buyer and seller agents (Sections 3.1 and 3.2, respectively). Then we briefly

when one operand is greater than the unit element and the other is smaller, we cannot expect to obtain a real compromise result using any of these six kinds of uninorm operator. However, because of (11) we can expect this through uninorm operator $\oplus_P$ and hence it is our choice (more details of which can be found in Sections 4.1 and 5).

deal with the communication aspects (Section 3.3). Finally, we specify the negotiation behaviour of our buyer and seller (Sections 3.4 and 3.5, respectively).

In what follows we adopt the CommonKADS framework [52] for our specification. This choice is based on our observation that human negotiations are heavily knowledge-based [11,56,57] and that the automated counterpart is therefore likely to be a knowledge intensive system. For such systems, CommonKADS is a reasonably common specification framework [52]. At this point, we make no claim that CommonKADS is any better than any of the other more common techniques for specifying agent systems; however it is also not provably worse. We use it as a tool because it is a well articulated framework with well-defined models that when taken together provide a comprehensive specification of a complex knowledge system. Moreover, we have found that the rigours of the framework are an important software engineering aid in ensuring that all the necessary details are present in our model and that our implementation from this specification was a reasonably straightforward endeavour.

3.1. Specifying the seller agent

The following represents our conceptualisation of a seller agent.

Definition 9. A *seller agent* is a 5-tuple knowledge system

$$(\mathcal{G}, \mathcal{A}, \mathcal{P}, \varpi, \Theta)$$

where:

(1) $\mathcal{G} = \{g_i \mid g_i = (c_i, r_i, u_i, p_i),\ p_i = (v_{i1}, \ldots, v_{in}),\ 0 \leqslant i \leqslant k\}$, called the *product model*, is the *domain knowledge* of the agent (i.e., the set of *products* the seller holds). c_i is the *restriction* attached to product g_i that a buyer agent must satisfy in order to obtain the product (e.g., buyers must be over 18 years old or the goods are for the export market only). The restriction is expressed as a Boolean expression (see Definition 8) of some primitive propositions. r_i is the *reward* associated with product g_i, which the seller agent may use to persuade a buyer agent to purchase the product. The reward is also expressed as a Boolean expression (see Definition 8) of some primitive propositions. u_i is the *profit* that the seller agent gets if product g_i is sold at a particular price. p_i, called the *product-attributes*, is the value vector of negotiable attributes (e.g., price, quality, model, volume, delivery date, expiry date, after-sale service and warranty) of product g_i. Finally, k is the total number of products the seller agent possesses.

(2) $\mathcal{A} = \{$generate, update, propose-restriction, propose-reward, receive, present$\}$ is the set of primitive actions that the seller agent can take during negotiations.
- Action generate puts forward a solution, from its product set, to satisfy the constraint set that a buyer has submitted. If more than one solution satisfies the constraint set, a solution with the highest profit is chosen in order to guarantee the maximum profit for the seller agent.
- Action update modifies (adds or deletes) constraints in a constraint set. The effect of deleting a constraint is to remove it from the set and then add its negation into the set.

– Action propose-restriction proposes the restriction attached to a product.
– Action propose-reward proposes the reward associated with a product.
– Action receive receives an offer from the buyer agent.
– Action present sends an offer to a buyer agent.

(3) $\mathcal{P}$ is the *behaviour protocol* that specifies the rules that the seller agent must obey during the course of the negotiation.

(4) $\varpi = (O_{seller}, O_{buyer})$ is the *communication port* of the seller agent; O_{seller} is a *seller's offer*, and O_{buyer} is a *buyer's offer*.[4]

(5) Θ = (**constraint-set**, **solution**, **last-solution**, **previous-solutions**) is the *working memory* of the seller agent. The **constraint-set** stores the *constraint*(s) that a buyer agent has so far submitted to the seller agent. The **solution** stores the current solution that the seller agent finds according to the **constraint-set** in the current round of negotiation. The **last-solution** stores the solution that the seller agent found in the last round of negotiation. The **previous-solutions** stores all the solutions that the seller agent found, before the current round, according to the same **constraint-set** the buyer had submitted.

These definitions mean that the seller agent consists of five main components: (1) the domain knowledge—a set of products plus their associated information, (2) a set of primitive actions, (3) a behaviour protocol, (4) the communication port, and (5) the working memory. This accords with the view that agents are action-based systems that can autonomously act upon the environment and/or interact with other agents [21].

The method used to generate offers, *the negotiation strategy*, is the way in which the actions generate, propose-restriction, and propose-reward work. In this context, action generate chooses the solution with the highest profit from the set of feasible solutions, and actions propose-restriction and propose-reward simply retrieve the attached restriction and the associated reward according to the name of a product.[5]

3.2. Specifying the buyer agent

Using the same representation scheme, the buyer agent can be viewed as consisting of five components: (1) a domain knowledge model consisting of the buyer's requirement model, the buyer's profile model, and the buyer's acceptability threshold, (2) a set of primitive actions, (3) a behaviour protocol, (4) a communication port, and (5) a working memory. More concretely:

Definition 10. A *buyer agent* is a 5-tuple knowledge system

$$(\mathcal{KD}, \mathcal{A}, \mathcal{P}, \varpi, \Theta)$$

where:

---

[4] Since the communication port involves both the buyer and seller agents, it is detailed after the buyer agent has been defined (see Section 3.3).

[5] The algorithms to implement these actions are straightforward, and so we omit their details for the sake of space.

(1) $\mathcal{KD} = (\mathcal{C}, \mathcal{B}, \tau)$ is the domain knowledge of the buyer agent.
   - $\mathcal{C} = (X, D, C^f, \rho, \varrho)$ is the *buyer's requirement model*. (X, D, C^f, ρ) is a PFCSP (see Definition 6). X is the set of attributes of the products. Each domain $d_i \in D$ is a set of possible values of an attribute of the products. C^f is a set of fuzzy constraints that express the buyer's requirements on the attributes of the desired product. Each constraint $R_i^f \in C^f$ is associated with a *priority* $\rho(R_i^f) \in [0, +\infty)$ as well as a *relaxing threshold* $\varrho(R_i^f) \in [0, 1]$. Given a cut level (see Definition 5), a fuzzy constraint can be *relaxed* if the cut level, by which the fuzzy constraint induces a crisp constraint, is not less than the relaxing threshold.
   - $\mathcal{B} = (F, t)$ is a fuzzy truth proposition system (see Definition 8), representing the *buyer's profile model*. This is the background information it uses to evaluate the seller's offer. $F = \{f_i \mid i = 1, \ldots, l\}$ is a set of fuzzy propositions. $t : F \to [0, 1]$ is a truth function.
   - $\tau \in [0, 1]$ is the *acceptability threshold* which the seller's offer must surpass to be acceptable for the buyer agent (see Definition 11).

(2) $\mathcal{A} = \{$select, verify, satisfy, evaluate, critque, relax, receive, present$\}$ is the set of *primitive actions* the buyer agent can take during negotiations.
   - Action select generates a crisp constraint induced from the fuzzy constraint with the highest priority in the buyer's constraint set. It does this by using the cut-set technique (see Definition 5) at cut level 1. If there is more than one constraint with the highest priority, choose randomly between them.
   - Action verify checks the attribute values of a seller offered product against the crisp constraints induced. It does this by using the cut-set technique (see Definition 5) at a given cut level, from the buyer's constraint set. If the test is passed, return "true"; otherwise, return "false".
   - Action satisfy calculates the overall satisfaction degree of all the fuzzy constraints whose induced crisp constraints have been submitted to the seller agent at the current point of the negotiation process.
   - Action evaluate evaluates a seller's offer to determine if it is acceptable. This offer consists of the attribute values of the product plus its attached restriction condition (if any) and associated reward (if any).
   - Action critique uses the cut-set technique (see Definition 5) to induce a crisp constraint from the fuzzy constraint with the highest priority among those whose induced crisp constraints are violated by the seller's offer.
   - Action relax relaxes the fuzzy constraint that has the lowest priority among the constraints that have already been submitted to the seller agent. The cut-set technique (see Definition 5) is used to relax a fuzzy constraint as little as possible. That is, first decrease the current cut level as little as possible (the formal definition for this will be given in Definition 16). Then, at the decreased cut level induce a crisp constraint from the fuzzy constraint. The crisp constraint is the relaxed constraint ready for submitting to the seller agent.
   - Action receive receives an offer from a seller agent.
   - Action present sends an offer to a seller agent.

(3) $\mathcal{P}$ is the *behaviour protocol* that specifies the rules that the buyer agent must obey during the course of the negotiation.

(4) $\varpi = (O_{seller}, O_{buyer})$ is the *communication port* of the buyer agent; O_{seller} is a *seller's offer*, and O_{buyer} is a *buyer's offer*.
(5) $\Theta =$ (**submitted-constraints**, **cut-level**) is the *working memory* of the buyer agent. The **submitted-constraints** stores the constraints that the buyer agent has so far sent to the seller. The **cut-level** stores the current degree to which the fuzzy constraints are relaxed.

As before, the buyer's negotiation strategy is defined by the way in which the actions select, critique, relax, evaluate and satisfy work. The operation of the first three actions is self-evident from the above definition. Those of the last two need to be further detailed. In our model, the buyer agent evaluates a seller's offer according to its acceptability for the offered product, which is defined as follows:

Definition 11. Suppose the acceptability threshold of a buyer agent is $\tau \in [0, 1]$. Let $\oplus$ be a uninorm operator with unit element τ which satisfies the following property:

$$\min\{a_1, a_2\} < \tau < \max\{a_1, a_2\} \quad \Rightarrow \quad \min\{a_1, a_2\} < a_1 \oplus a_2 < \max\{a_1, a_2\}. \tag{15}$$

Let $(X, D, C^f, \rho, \varrho)$ be the requirement model (see Definition 10), and (F, t) be the buyer's profile model (see Definition 10). For an offered product

$$g_i = (c_i, r_i, p_i),$$

where:

- c_i is the restriction expressed as a Boolean expression of $f'_1, \dots, f'_s \in F$;
- r_i is the reward expressed as a Boolean expression of $f''_1, \dots, f''_t \in F$; and
- $p_i = (v_{i1}, \dots, v_{in})$ is the value vector of the product's attributes.

Now the buyer's *acceptability* can be expressed as:

$$\mathit{acceptability}(g_i) = \min\bigl\{\alpha(v_{i1}, \dots, v_{in}), \beta\bigr\} \oplus \bigl((1-\tau)\gamma + \tau\bigr), \tag{16}$$

where:

$$\alpha(v_{i1}, \dots, v_{in}) = \min\left\{\left(\frac{\rho(R^f)}{\rho_{\max}}\right) \diamond \mu_{R^f}(v_{var(R^f)}) \mid R^f \in C^f\right\}, \tag{17}$$

$$\beta = 1 - c'_i\bigl(t(f'_1), \dots, t(f'_s)\bigr), \tag{18}$$

$$\gamma = b'_i\bigl(t(f''_1), \dots, t(f''_t)\bigr). \tag{19}$$

The meaning of the three parameters in acceptability formula (16) is as follows:

(1) $\alpha(v_{i1}, \dots, v_{in}) \in [0, 1]$ (α for short) is the overall satisfaction degree to which the compound label $(v_{i1}, \dots, v_{in})$ on attributes of product g_i satisfies all the buyer agent's constraints. Actually, α is calculated from the buyer's requirement model by using the overall satisfaction degree formula (7). In what follows, α is called the *requirement satisfiability* of the product.

(2) $\beta \in [0,1]$ is the possibility the buyer agent can obey the attached restriction of product g_i (hence it is called the product's *restriction obedience*). Restricting c_i for the buyer means that c_i should not be true, to some extent, for the buyer. So, the degree to which the buyer can obey restriction c_i is $1 - c_i'(t(f_1'), \ldots, t(f_s'))$ since $c_i'(t(f_1'), \ldots, t(f_t'))$ represents the degree to which c_i is true for it. Actually, β is calculated from the buyer's profile model by using fuzzy truth formulas (12), (13) and (14).
(3) $\gamma \in [0,1]$ is the degree to which the reward associated with product g_i is valuable (hence it is called the product's *reward value*). Actually, γ is also calculated from the buyer's profile model by using the fuzzy truth formulas (12), (13) and (14).

The rationale for acceptability formula (16) will be given in Section 4 where we concentrate on various properties of our model. Notice, however, that since the uninorm operator is closed on [0, 1] and the two operands in acceptability formula (16) are in [0, 1], the buyer's acceptability for a product takes on a value in the range [0, 1]. The larger the value, the more acceptable the product. In particular, when it takes value 1, the product is completely acceptable; when it takes value 0, the product is absolutely unacceptable. In between it is more or less acceptable.

Finally, the definition below gives the formula used by action satisfy to calculate the buyer's *potential payoff*[6] during negotiation.

Definition 12. Suppose that in a given round of negotiation the crisp constraints the buyer agent has submitted to the seller agent are $R_1'^c, \ldots, R_m'^c$ and the fuzzy constraints which induced these crisp constraints are $R_1'^f, \ldots, R_m'^f$. The buyer's potential payoff with respect to $R_1'^f, \ldots, R_m'^f$, denoted as $pp(R_1'^f, \ldots, R_m'^f)$, is the overall satisfaction degree of $R_1'^f, \ldots, R_m'^f$ for a product that the seller agent offers, when the attributes' values $av_1, \ldots, av_l$ of the product satisfy $R_1'^c, \ldots, R_m'^c$. That is,

$$pp\big(R_1'^f, \ldots, R_m'^f\big) = \min\left\{ \frac{\rho(R_i'^f)}{\rho_{\max}} \diamond \mu_{R_i'^f}(\vec{a}) \mid \mu_{R_i'^c}(\vec{a}) = 1,\ 1 \leqslant i \leqslant m \right\}, \tag{20}$$

where $\vec{a} = (av_1, \ldots, av_l)$ and $\rho_{\max} = \max\{\rho(R_1'^f), \ldots, \rho(R_m'^f)\}$.

Our definition of the potential payoff conforms to the following: if in round k of the negotiation the buyer has submitted crisp constraints $R_1'^c, \ldots, R_m'^c$ that are induced at this cut level from $R_1'^f, \ldots, R_m'^f$, and in round $k+1$ the seller's offer to the buyer is a product satisfying $R_1'^c, \ldots, R_m'^c$, then the overall satisfaction degree of $R_1'^f, \ldots, R_m'^f$ for the product is the buyer's payoff that it may get in round k. In other words, the concept of potential payoff is used to ensure that the buyer's payoff will not be unnecessarily lost (i.e., when

[6] Intuitively, this can be viewed as the payoff (satisfaction degree of its constraints on the desired product) that the buyer agent expects to receive when submitting its constraint at a given point in the negotiation process. That is, in the next round if the seller's offer satisfies the buyer's constraints, the buyer's potential payoff becomes its actual payoff. This concept of potential payoff is different from the standard concept of expected utility [22,40] that represents a kind of "average payoff" of various uncertain consequences (if a particular consequence really occurs, the actual payoff is the utility of the consequence).

the buyer offers new constraints its potential payoff should not decrease). In Section 4, we will show the design indeed achieves this.

3.3. Inter-agent communication

The agents communicate using an alternating offers protocol [48]. The communication ports of our seller and buyer agents consist of a pair of messages the two agents present to one another. As is common in the field [59], we separate the representation of messages between agents into two levels—the *communication language* level and the *content language* level.

(1) At the *communication language* level, we adopt the KQML standard [10] with several minor additions. Specifically, we provide performatives through which the seller agent can ask the buyer agent to relax constraints, and check the product against its requirement and profile models; and through which the buyer agent can ask the seller agent to find or refind a product compatible with the submitted constraints, and ask the seller agent to end negotiation. The details of these performatives will be given in Definitions 13 and 14 below.
(2) At the *content language* level, a seller's offer is a solution (a product specified by a compound label on the attributes of products) to a CSP, plus any associated information (its attached restriction condition or associated reward). Given the fact that the message content of a buyer's offer contains constraints, we choose to base our content language on the Constraint Choice Language [59] since this is specifically designed for carrying, between agents, constraints or solutions to CSPs. This language needs a minor extension for our purposes in order for it to have the ability to contain information about the attached restriction and associated reward of a product.[7]

Formally, and more precisely, we have:

Definition 13. A *seller's offer* is a message with the following structure:

$$O_{seller} = (product, restriction, reward, performative),$$

where:

(1) The item **product** $= (a_1, \ldots, a_n)$ is a *value vector* of attributes of the *product* that the seller agent believes satisfies the constraints submitted by the buyer agent.
(2) The item **restriction** attached to the product is a Boolean expression (see Definition 8).
(3) The item **reward** associated with the product is a Boolean expression (see Definition 8).

[7] In this work, we assume that the agents share a common ontology for the terms they exchange at this level. While this is obviously a simplification, it is one that we believe is reasonable in this context given the aims of this paper.

(4) The item **performative** represents the action that the seller agent wants the buyer agent to take after receiving the message:[8]
   - `check`: The seller agent is asking the buyer agent to check whether the product is acceptable. That is, to check whether the product plus any accompanying information can satisfy its whole constraint set (because some of the buyer's constraints may not yet have been revealed to the seller) and whether the buyer's minimum interest (its acceptability threshold) can be guaranteed.
   - `relax`: The seller agent is asking the buyer agent to relax one of the constraints it has submitted. This is sent when the seller agent cannot find a product that satisfies the constraints submitted by the buyer agent.

The selection of these performatives is based on the following intuitions. Firstly, since negotiation involves mutual agreement by both parties, if the seller finds a product that it believes satisfies the buyer's requirements (as so far specified) then what it would like to do is to ask the buyer to `check` whether the product does indeed meet all the constraints. Secondly, sellers always prefer to make a (profitable) sale. So, even if sellers cannot find products that satisfy the constraints submitted by buyers, they are unlikely to say "`fail`". Rather they prefer to ask the buyers to `relax` their submitted constraints to see if a compatible product can be found.

Definition 14. A *buyer's offer* is a message with the following structure:

$$O_{buyer} = (constraint, performative),$$

where:

(1) The item **constraint** contains the constraints that the buyer agent uses to specify the product it wants to buy. In particular, if the buyer agent wants to relax a constraint c that it has already submitted, it is represented as $-c$ and $+c'$, where c' is the relaxed constraint. If the buyer agent wants to add a new constraint c, the constraint is represented as $+c$.
(2) The item **performative** represents the action that the buyer agent wants the seller agent to take after receiving the message:
   - `find`: The buyer agent is asking the seller agent to find a product that satisfies the constraints it has submitted.
   - `refind`: The buyer agent is asking the seller agent to find an alternative product for it. This is sent when the buyer cannot accept the product that it has just been offered.
   - `deal`: The buyer agent is asking the seller agent to end their negotiation. It is sent when a successful deal has been made.

[8] Naturally, since the agents are autonomous these requests do not have to be adhered to. However, since it is in the interests of both agents to do so, we assume in our subsequent descriptions that the agents do indeed perform the actions suggested by their counterpart.

– `fail`: The buyer agent is asking the seller agent to end their negotiation. It is sent when the seller agent cannot offer a product compatible with the buyer's submitted constraints and the buyer cannot relax any constraint further.

The selection of the performatives is based on the following intuitions. Firstly, when a buyer submits a constraint to a seller, the buyer's intention is to ask the seller to `find` a product satisfying the present constraint and any previously submitted ones. Secondly, when a buyer reviews a seller's offer which does not violate any of its constraints at the current cut level, but does not reach its acceptability threshold, the buyer does not want to give up on the chance of reaching a deal. Thus, it asks the seller to `refind` an alternative. Thirdly, the buyer needs a means of signifying the end of a negotiation (whether a deal is made or not).

3.4. The seller agent's behaviour

After specifying the communication between the buyer and the seller, we turn to their behaviour. This subsection specifies the seller agent's behaviour protocol, and the next subsection specifies the buyer's.

The behaviour protocol of a seller agent is shown in Fig. 1. The basic assumption behind it is that the seller agent is trying to obtain the best deal it can. Therefore, according to the tenets of the principled negotiation approach when the buyer cannot accept the seller's offer, the seller always tries to find a trade-off offer with the same profit to it as the previous offer. Only when the seller cannot do this, does it makes some concession (i.e., provides another offer with less profit for it). However, the seller always wants to see if it can reach a profitable agreement since it receives no revenue unless it make agreements. Thus if the buyer agent does not accept its offer (even with the inducement of a reward) and it cannot find an alternative offer, it will ask the buyer to relax its constraints.

Therefore, during the course of a negotiation, the seller acts according to the messages it receives as follows:

(1) `find` or `refind` (as shown on lines 4–25 in Fig. 1): First the seller agent tries to generate a solution for the buyer agent.
- In the case that the performative of the buyer message is `find` (as shown on lines 5–7 in Fig. 1): the constraints contained in the message are added in and/or deleted from its constraint set. Then, it tries to find the solution (product) that is consistent with its constraint set and at the same time maximises its own profit.
- In the case that the performative of the buyer message is `refind` (as shown on lines 8–9 in Fig. 1): its constraint set remains unchanged, but it tries to generate a new solution to the constraint set, while still trying to maximise its own profit.

For either of the above two cases:
- In the case that the seller agent succeeds in finding a product (as shown on lines 11–14 in Fig. 1), it will ask the buyer agent to `check` whether the product satisfies all its constraints (not all of them may have been exposed to the seller) and whether the product reaches its acceptability threshold.

```
1.  last-solution := NIL; constraint-set := ∅;
2.  REPEAT
3.      receive(O_buyer);
4.      IF O_buyer.performative = ‘find’ OR ‘refind’ THEN
5.        IF O_buyer.performative = ‘find’ THEN
6.            constraint-set := update(constraint-set,O_buyer.constraint);
7.            solution := generate(constraint-set); previous-solutions := ∅;
8.        ELSE
9.            solution := generate(constraint-set ∪ {solution ∉ previous-solutions});
10.       END IF;
11.       IF solution ≠ NIL THEN
12.           O_seller.product := solution; O_seller.reward := NIL;
13.           O_seller.restriction := propose-restriction(solution.product);
14.           O_seller.performative := ‘check’
15.       ELSE IF last-solution = NIL THEN
16.           O_seller.performative := ‘relax’;
17.       ELSE IF propose-reward(last-solution.product) ≠ NIL
18.                AND last-solution.reward = NIL THEN
19.           O_seller.reward := propose-reward(last-solution.product);
20.           O_seller.product := last-solution.product;
21.           O_seller.restriction := last-solution.restriction;
22.           O_seller.performative := ‘check’
23.       ELSE O_seller.performative := ‘relax’;
24.       END IF; END IF; END IF;
25.     END IF;
26.     IF O_buyer.performative = ‘deal’ THEN decision := O_seller END IF;
27.     IF O_buyer.performative = ‘fail’ THEN decision:=NIL END IF;
28.     present(O_seller); last-solution := O_seller;
29.     previous-solutions := previous-solutions ∪ {last-solution};
30. UNTIL O_buyer.performative = ‘fail’ OR ‘deal’ END REPEAT;
31. RETURN decision
```

Fig. 1. The specification of the seller agent’s behaviour. The explanation of the symbols is given in Definitions 9, 13 and 14.

– In the case that the seller agent fails to find a product that satisfies the buyer’s constraints (as shown on lines 15–23 in Fig. 1), the next action depends on the negotiation context. If the seller agent is in the first round of negotiation, i.e., the last solution does not exist (as shown on lines 15–16 in Fig. 1), it will ask the buyer agent to `relax` its constraint(s). Otherwise (as shown on lines 17–23 in Fig. 1), the seller agent insists on the last solution (i.e., the product it offered to the buyer agent in the last round of negotiation) if the reward associated with the last offer has not been mentioned before (as shown on lines 17–22 in Fig. 1), but offers some reward to see whether the buyer’s acceptability for the product can be increased to such a degree that the product becomes acceptable. If the reward was already mentioned to the buyer before (as shown on line 23 in Fig. 1), the seller asks the buyer to relax its constraint.

(2) `deal` or `fail` (as shown on lines 26, 27 and 30 in Fig. 1): The negotiation has either succeeded or failed and thus the negotiation process terminates.

The above protocol defines the rules that a seller agent should obey. In addition to this, however, the other component that needs to accompany the behaviour protocol is the negotiation strategy. The strategy is the method that the seller agent uses to maximise its own payoff within the confines of the behaviour protocol. In this case, the strategy of our seller agent works in the following way:

(1) When the seller agent generates a solution for a CSP, if there are multiple alternatives it always chooses the one from which it gains the highest profit. It has, after all, a degree of self-interest and so tries to maximise its profit.
(2) When a buyer agent asks the seller agent to `refind` an alternative, the seller agent uses action generate to provide a new solution if it can. This is equivalent to always trying to make a trade-off among the product's different attributes since action generate always provides the solution with the highest profit if one exists. If the highest profit is the same as that of the seller's last offer, this keeps the seller's profit unchanged and so the solution represents a trade-off. Here in the case of the buyer asking for an alternative, we do not choose a strategy of only reducing price to reach a deal. The reason for this is that according to the principled approach to negotiation only reducing price, but not obtaining compensation on some other negotiation issues, is not a prudent course of action. Thus in our model we do not reduce the price of the same product, but we offer an alternative product (deal). This alternative may well have a lower price but will necessarily be better for the seller (but worse for the buyer) on some other attributes.
(3) When the seller agent cannot find a new solution, it always tries to provide the buyer agent with some additional reward. In this way, the seller agent tries to keep its profit largely unchanged (assuming the cost of providing the reward is small in comparison to the cost of the product). Notice that the seller agent does not provide the buyer agent with a reward when it can find a solution. It only offers rewards if the buyer agent cannot accept a solution and the seller agent cannot find an alternative. The goal of such a design is to satisfy the buyer as much as possible. By doing so the seller aims to build or maintain its reputation and so make more profit in the long term.

3.5. The buyer agent's behaviour

The behaviour protocol of a buyer agent is shown in Fig. 2. The basic assumption behind it is that the buyer agent tries to maximise the overall satisfaction degree of its requirements (constraints). Accordingly, for the buyer we need to ensure that when it makes a concession (i.e., it relaxes a constraint), it should relax a less important constraint; when it submits constraints, it submits more important ones before less important ones. Meanwhile we assume that it also always prefers a (profitable) deal to be struck. Hence, if the seller agent cannot offer it a product that satisfies the constraints that it has submitted, it will endeavour to relax some of its constraints to a certain extent; if the buyer can accept the seller's offer, it does not lie to the seller hoping it will make more concessions or offering it more benefits

1. O_{buyer}.performative := ‘find’; O_{buyer}.constraint := select;
2. present(O_{buyer}); submitted-constraint-set := ∅;
3. REPEAT
4. receive(O_{seller});
5. submitted-constraints := submitted-constraints ∪ {O_{buyer}.constraint};
6. IF O_{seller}.performative = ‘check’ THEN
7. cut-level := satisfy(submitted-constraint-set,O_{buyer}.product);
8. IF verify(O_{seller}.product, cut-level) = true THEN
9. IF evaluate(O_{seller}) = true THEN
10. O_{buyer}.performative := ‘deal’;
11. decision := O_{seller};
12. ELSE
13. O_{buyer}.performative := ‘refind’
14. END IF;
15. ELSE
16. O_{buyer}.constraint := critique(O_{seller}, cut-level);
17. O_{buyer}.performative := ‘find’;
18. END IF;
19. END IF;
20. IF O_{seller}.performative = ‘relax’ THEN
21. (O_{buyer}.constraint, cut-level) := relax(submitted-constraints, cut-level);
22. IF O_{buyer}.constraint ≠ NIL THEN
23. O_{buyer}.performative := ‘find’
24. ELSE
25. O_{buyer}.performative := ‘fail’; decision := NIL;
26. END IF;
27. END IF;
28. present(O_{buyer});
29. UNTIL O_{buyer}.performative = ‘fail’ OR ‘deal’ END REPEAT;
30. RETURN decision

Fig. 2. The specification of the buyer agent’s behaviour. The explanation of the symbols used is given in Definitions 10, 13 and 14.

since this jeopardises the deal. A secondary objective of the buyer agent is to minimise the amount of information that it has to reveal in order to strike a deal (as per the discussion in Section 1).

Similar to the seller agent, the buyer agent’s actions are driven by the messages it receives. As shown on line 1 in Fig. 2, the buyer agent initiates the negotiation by selecting the constraint with the highest priority from its constraint set and submitting this constraint to the seller agent.[9] Then, in each subsequent round of negotiation the buyer agent acts according to the performative message the seller sends to it:

[9] Intuitively, the constraint with the highest priority should be satisfied before any other. Formally, Theorem 3 (given in the next section) guarantees that making such a choice maximises the buyer’s overall satisfaction degree.

(1) `check` (as shown on lines 6–19 in Fig. 2): The buyer agent checks whether the product offered by the seller agent satisfies all its requirement constraints and whether its acceptability threshold for the product is reached. Notice that if a seller's offer violates a fuzzy constraint this means that it violates the crisp constraint induced by the fuzzy constraint at the current cut level.
   - If the seller's offer is acceptable (i.e., there are no violated constraints and the buyer's acceptability for the product is above the acceptability threshold) (as shown on lines 8–11 in Fig. 2), the buyer agent indicates that a deal is made. Thus, the negotiation procedure terminates.
   - If the seller's offer is unacceptable (i.e., there are no violated constraints but the buyer's acceptability for the product is less than the buyer's acceptability threshold) (as shown on lines 12–13 in Fig. 2), the buyer agent asks the seller agent to refind an alternative offer.
   - If the seller's offer satisfies some of the constraints, but violates others (as shown on lines 15–17 in Fig. 2), the buyer agent will pick out the constraint with the highest priority from the constraints that are violated, and ask the seller agent to refind an alternative offer.

(2) `relax` (as shown on lines 20–27 in Fig. 2): The buyer agent tries to relax the constraints it has submitted to the seller to see if a deal can be made.
   - If there is a constraint which can be relaxed (i.e., after the current cut level is decreased, it is still not less than its relaxing threshold) (as shown on lines 22–23 in Fig. 2), the buyer agent will ask the seller to delete the constraint to be relaxed, and add the newly relaxed one. Meanwhile, the current cut level is updated to the newly decreased one.
   - If there are no constraints that can be relaxed (i.e., after the current cut level is decreased, it becomes less than its relaxing threshold) (as shown on lines 24–25 in Fig. 2), the buyer agent will inform the seller agent that their negotiation has failed. Thus, the negotiation process terminates.

Let us explain why, when the buyer agent receives a product offer, the `check` is necessary. This is because in each round of their negotiation the buyer agent submits only one new constraint or relaxes an already submitted one. Thus, during negotiation it is often the case that there are some constraints that the buyer agent has not revealed to the seller agent. The rationale of this design is that the buyer and seller agents are in a semi-competitive situation in which information revelation can be exploited. On the other hand, if an offered product satisfies all the buyer agent's constraints it is unnecessary for the buyer agent to expose all of them to the seller agent. Thus there is clearly a trade-off to be made. In any case, our incremental protocol ensures that the buyer agent minimises the amount of information that it reveals during the course of the negotiation. We will return to this point in Sections 4 and 5.

In sum, the strategy that the buyer agent employs to maximise the overall satisfaction degree of its requirements (constraints) can be stated as follows:

(1) When submitting constraints, always use a cut level as high as possible to transform a fuzzy constraint to a hard (crisp) one and then submit the hard constraint to the seller agent.
(2) When submitting a new constraint to the seller agent, choose the highest priority one among those not yet submitted.
(3) When relaxing a constraint, always try the one with the lowest priority first. If there is more than one, choose randomly. If a constraint with the lowest priority cannot be relaxed, then try the one with the second lowest priority, and so on.

The aim of such a design is to guarantee that the buyer's overall satisfaction degree is maximised. In the next section we will formally prove our design indeed achieves this goal.

4. Properties of the negotiation model

This section analyses the key aspects and properties of our negotiation model. Particular attention is focused on the condition under which a buyer accepts a deal (Section 4.1), the foundational principles of the buyer's negotiation strategy (Section 4.2) and the overall properties of the negotiation outcomes and the revelation of information that occurs when using our model (Section 4.3).

4.1. The buyer's acceptability conditions

The design of the acceptability formula (16) is based on the following intuitions. Firstly, the product must more or less satisfy the buyer's requirement constraints, and the buyer must more or less satisfy the product's attached restrictions. In other words, when the product's requirement satisfiability $\alpha = 0$ or the restriction obedience $\beta = 0$, the product is absolutely unacceptable. Secondly, when the product's requirement satisfiability α, restriction obedience β and reward value γ increase, the buyer's acceptability for a product should also increase. Thirdly, a reward is used to try and increase the buyer's acceptability for a product. Fourthly, if the buyer agent does not care at all about the reward associated with a product (or equally there is no reward), the reward should not increase the buyer's acceptability for the product. Finally, when a seller uses a reward to try to raise a buyer's acceptability, if the main factor, the product's requirement satisfiability α, is originally less than the acceptability threshold, the reward as the secondary factor can raise the acceptability to some extent, but this raise should not make the product maximally acceptable. Formally, we have the following definition:

Definition 15. The function $f : [0,1] \times [0,1] \times [0,1] \to [0,1]$ is the *buyer's acceptability function* for a product if it satisfies:

(1) $\forall \alpha, \beta, \gamma \in [0,1]$, $f(0, \beta, \gamma) = 0$, $f(\alpha, 0, \gamma) = 0$;
(2) $\forall \alpha, \alpha', \beta, \beta', \gamma, \gamma' \in [0,1], \alpha \geqslant \alpha', \beta \geqslant \beta', \gamma \geqslant \gamma' \Rightarrow f(\alpha, \beta, \gamma) \geqslant f(\alpha', \beta', \gamma')$;
(3) $\forall \alpha, \beta, \gamma \in [0,1]$, $f(\alpha, \beta, \gamma) \geqslant \min\{\alpha, \beta\}$;
(4) $\forall \alpha, \beta \in [0,1]$, $f(\alpha, \beta, 0) = \min\{\alpha, \beta\}$; and
(5) $\forall \alpha, \beta, \gamma \in [0,1], \alpha < \tau, \gamma > 0 \Rightarrow \alpha < f(\alpha, \beta, \gamma) < 1$.

The following theorem shows that all five axioms listed in the above definition are indeed met in formula (16).

Theorem 1. *Formula* (16) *is appropriate for using as a buyer's acceptability function.*

Proof. Let $f(\alpha,\beta,\gamma)$ be *acceptability*(g_i) given by (16). In the following, we show it satisfies all the axioms in Definition 15:

(1) By Definition 11 and property (4) in Lemma 1, we have:

$$f(0,\beta,\gamma)=\min\{0,\beta\}\oplus\big((1-\tau)\gamma+\tau\big)=0\oplus\big((1-\tau)\gamma+\tau\big)=0,$$
$$f(\alpha,0,\gamma)=\min\{\alpha,0\}\oplus\big((1-\tau)\gamma+\tau\big)=0\oplus\big((1-\tau)\gamma+\tau\big)=0.$$

(2) Since *min* and uninorm operator $\oplus$ (see Definition 7) are increasing operators, axiom (2) in Definition 15 is satisfied.

(3) First notice that $(1-\tau)\gamma+\tau\geqslant\tau$. Thus, if $\min\{\alpha,\beta\}\geqslant\tau$, by properties (1) and (3) of Lemma 1 we have:

$$\begin{aligned}f(\alpha,\beta,\gamma)&=\min\{\alpha,\beta\}\oplus\big((1-\tau)\gamma+\tau\big)\\&\geqslant\max\big\{\min\{\alpha,\beta\},(1-\tau)\gamma+\tau\big\}\\&\geqslant\min\{\alpha,\beta\}.\end{aligned}$$

(4) By Definitions 7 and 11, we have:

$$f(\alpha,\beta,0)=\min\{\alpha,\beta\}\oplus\big((1-\tau)\times 0+\tau\big)=\min\{\alpha,\beta\}\oplus\tau=\min\{\alpha,\beta\}.$$

(5) Since $\alpha<\tau$ implies $\min\{\alpha,\beta\}<\tau$ and $\gamma>0$ implies $((1-\tau)\gamma+\tau)>\tau$, by assumption (15) for the acceptability formula (16) we have:

$$\alpha\leqslant\min\{\alpha,\beta\}<\min\{\alpha,\beta\}\oplus\big((1-\tau)\gamma+\tau\big)<(1-\tau)\gamma+\tau\leqslant 1.$$

That is, $\alpha<f(\alpha,\beta,\gamma)<1$. □

Actually, formula (16) is just an instantiation of the axiomatic definition (Definition 15) of the buyer's acceptability function.[10] There are likely to be other instantiations, but this issue is beyond the scope of this paper.

In order to actually use acceptability formula (16) to calculate the buyer's acceptability for a product, we need to instantiate the uninorm operator and the priority operators for calculating the satisfiability. The condition that the uninorm operator should meet is specified, i.e., (15). By Lemma 2, the uninorm operator $\oplus_P$ given by (11) meets (15) because of (10), and so $\oplus_P$ can be employed in acceptability formula (16).

[10] Generally speaking, in engineering any system, it is important to first define the requirements and then pick an operator to fit. In defining the buyer's acceptability for a product, what we first find are these requirements (i.e., the axioms listed in Definition 15), then in order to fit these requirements we chose a uninorm in formula (16). In this paper, however, for expository purposes we present these in the inverse order (i.e., we recall the uninorm operator first, then introduce the buyer's acceptability function (16), and finally present the requirements for the acceptability).

Now we turn to the issue of instantiating the priority operator in order to instantiate formula (17). Intuitively, we wish that the result obtained when things are prioritised is different from the one obtained when things are not. Thus:

Definition 16. A priority operator $\circ$ is a regular priority operator if:

$$\forall a_1, a_2 \in [0,1), \quad a_1 \circ a_2 \neq a_2. \tag{21}$$

Within this class of operators, we use the following specific one:

Theorem 2. *Operator* $\circ : [0,1] \times [0,1] \to [0,1]$*, defined as follows, is a priority operator*:

$$a_1 \circ a_2 = (a_2 - 1)a_1 + 1. \tag{22}$$

Proof. First, we prove operator $\circ$, given by (22), is a priority operator. In fact, the axioms of priority operators as listed in Definition 6 are satisfied with the operator as shown in the following:

$$a_2 \leqslant a_2' \quad \Rightarrow \quad (a_2 - 1)a_1 + 1 \leqslant (a_2' - 1)a_1 + 1 \quad \Rightarrow \quad a_1 \circ a_2 \leqslant a_1 \circ a_2',$$
$$a_1 \leqslant a_1' \quad \Rightarrow \quad (a_2 - 1)a_1 + 1 \geqslant (a_2 - 1)a_1' + 1 \quad \Rightarrow \quad a_1 \circ a_2 \geqslant a_1' \circ a_2,$$
$$1 \circ a_2 = (a_2 - 1)a_1 + 1 = (a_2 - 1) \times 1 + 1 = a_2,$$
$$0 \circ a_2 = (a_2 - 1)a_1 + 1 = (a_2 - 1) \times 0 + 1 = 1.$$

Then we prove the priority operator is also a regular priority operator (i.e., to check whether (21) is satisfied as well). Now since

$$(a_2 - 1)a_1 + 1 = a_2 \quad \Leftrightarrow \quad (a_2 - 1)(a_1 - 1) = 0 \quad \Leftrightarrow \quad a_1 = 1 \vee a_2 = 1,$$

(21) holds. $\square$

The above theorem guarantees that the priority operator given by (22) satisfies our assumption that the result with priority is not the same as the one without priority. For example, in (17) let $\mu_{R_1^f}(v_{var(R_1^f)}) = 0.8$ and $\rho(R_1^f) = 10$, $\mu_{R_2^f}(v_{var(R_2^f)}) = 0.9$ and $\rho(R_2^f) = 9$, $\mu_{R_3^f}(v_{var(R_3^f)}) = 0.6$ and $\rho(R_3^f) = 8$, and $\mu_{R_4^f}(v_{var(R_4^f)}) = 0.65$ and $\rho(R_4^f) = 7$, then by using (22) as the instantiation of the priority operator in (17), we obtain:

$$\begin{aligned}
\alpha = \min\Bigg\{ &\big(\mu_{R_1^f}(v_{var(R_1^f)}) - 1\big) \times \frac{\rho(R_1^f)}{\max\{\rho(R_i^f) \mid i = 1, \ldots, 4\}} + 1, \\
&\big(\mu_{R_2^f}(v_{var(R_2^f)}) - 1\big) \times \frac{\rho(R_2^f)}{\max\{\rho(R_i^f) \mid i = 1, \ldots, 4\}} + 1, \\
&\big(\mu_{R_3^f}(v_{var(R_3^f)}) - 1\big) \times \frac{\rho(R_3^f)}{\max\{\rho(R_i^f) \mid i = 1, \ldots, 4\}} + 1, \\
&\big(\mu_{R_4^f}(v_{var(R_4^f)}) - 1\big) \times \frac{\rho(R_4^f)}{\max\{\rho(R_i^f) \mid i = 1, \ldots, 4\}} + 1 \Bigg\}
\end{aligned}$$

$$= \min\left\{(0.8-1)\times\frac{10}{10}+1, (0.9-1)\times\frac{9}{10}+1,\right.$$
$$\left.(0.6-1)\times\frac{8}{10}+1, (0.65-1)\times\frac{7}{10}+1\right\}$$
$$= 0.68 \neq \min\{0.8, 0.9, 0.6, 0.65\}.$$

So, when we calculate the buyer's satisfiability by (17), we employ priority operator (22).

4.2. The buyer's negotiation strategy

One of key parts of the buyer's model is how it implements the relax action (see Section 3.2). Here the strategy used is based on the following theorem:

Theorem 3. *In a PFCSP (X, D, C^f, ρ), for $R_i^f, R_j^f \in C^f$, suppose $\rho(R_i^f) \geqslant \rho(R_j^f)$, and there are two different compound labels v_X and v'_X such that $\forall R^f \in C^f$,*

(1) *when $R^f \neq R_i^f$ and $R^f \neq R_j^f$, $\mu_{R^f}(v_{var(R^f)}) = \mu_{R^f}(v'_{var(R^f)})$,*
(2) *when $R^f = R_i^f$, $\mu_{R^f}(v_{var(R^f)}) = \mu_{R^f}(v'_{var(R^f)}) + \delta$,*
(3) *when $R^f = R_j^f$, $\mu_{R^f}(v'_{var(R^f)}) = \mu_{R^f}(v_{var(R^f)}) + \delta$,*

where $\delta \geqslant 0$. If

$$\frac{\rho(R_i^f)}{\rho_{\max}} \diamond \mu_{R_i^f}(v_{var(R_i^f)}) \leqslant \frac{\rho(R_j^f)}{\rho_{\max}} \diamond \mu_{R_j^f}(v_{var(R_j^f)}), \tag{23}$$

then

$$\alpha_\rho(v_X) \geqslant \alpha_\rho(v'_X). \tag{24}$$

Proof. Since the priority operators are increasing with respect to their second operand, from assumptions (2) and (3) of the theorem, we have:

$$\frac{\rho(R_i^f)}{\rho_{\max}} \diamond \mu_{R_i^f}(v_{var(R_i^f)}) \geqslant \frac{\rho(R_i^f)}{\rho_{\max}} \diamond \mu_{R_i^f}(v'_{var(R_i^f)}), \tag{25}$$

$$\frac{\rho(R_j^f)}{\rho_{\max}} \diamond \mu_{R_j^f}(v_{var(R_j^f)}) \leqslant \frac{\rho(R_j^f)}{\rho_{\max}} \diamond \mu_{R_j^f}(v'_{var(R_j^f)}). \tag{26}$$

Thus, by (23), we have:

$$\frac{\rho(R_i^f)}{\rho_{\max}} \diamond \mu_{R_i^f}(v'_{var(R_i^f)}) \leqslant \frac{\rho(R_j^f)}{\rho_{\max}} \diamond \mu_{R_j^f}(v'_{var(R_j^f)}). \tag{27}$$

Since the minimum operator is used in the overall satisfaction degree formula (7), because of (23) and (27) the following two values

$$\frac{\rho(R_j^f)}{\rho_{\max}} \diamond \mu_{R_j^f}(v_{var(R_j^f)}), \quad \frac{\rho(R_j^f)}{\rho_{\max}} \diamond \mu_{R_j^f}(v'_{var(R_j^f)})$$

are independent of the order between the overall satisfaction degrees, $\alpha_\rho(v_X)$ and $\alpha_\rho(v'_X)$, for v_X and v'_X. Thus, noticing assumption (1) of the theorem, by the overall satisfaction degree formula (7), the order between $\alpha_\rho(v_X)$ and $\alpha_\rho(v'_X)$ depends only on the order between:

$$\frac{\rho(R_i^f)}{\rho_{\max}} \diamond \mu_{R_i^f}(v_{var(R_i^f)}), \quad \frac{\rho(R_i^f)}{\rho_{\max}} \diamond \mu_{R_i^f}(v'_{var(R_i^f)}).$$

Therefore, by the order of (25) and considering that the minimum operator used in the overall satisfaction degree formula (7) is increasing with respect to its individual operands, (24) holds. □

Intuitively speaking, the theorem reveals that if an agent wants to raise the overall satisfaction degree of all its prioritised constraints, a constraint with a relatively high priority should be more sufficiently satisfied than a constraint with a relatively low priority. In our case, the buyer agent always tries to maximise the overall satisfaction degree of all its prioritised constraints since it has a degree of self-interest. As a result, the strategy used in the buyer's action relax is to relax the constraint that has the lowest priority.

For similar reasons, the strategy used in the buyer's action select (specified in Definition 10) is to submit to the seller agent the constraint with the highest priority among all its constraints; the strategy used in action critique (specified in Definition 10) is to submit to the seller agent the constraint with the highest priority among those that are violated. The second point to note about the strategy used in the buyer's critique action is that the newly submitted constraint should, wherever possible, keep the buyer's potential payoff unchanged. The following theorem guarantees this point is indeed realised. Before giving the theorem, we first give a lemma about priority operators.

Lemma 3. *Suppose operator $\diamond$ satisfies axioms* (2) *and* (3) *of priority operators which are listed in Definition* 6. *Then*:

$$a_1 \diamond a_2 \geqslant a_2. \tag{28}$$

Proof. By axioms (2) and (3) of the priority operators (see Definition 6), we have:

$$a_1 \diamond a_2 \geqslant 1 \diamond a_2 = a_2. \qquad \square$$

Theorem 4. *In the kth round of negotiation, let the buyer's potential payoff be $pp(R_1'^f, \ldots, R_m'^f)$ (given by* (20)*). If in round $k+1$ the buyer agent takes action* critique *and submits to the seller agent a crisp constraint $R_{m+1}'^c$ induced by fuzzy constraint $R_{m+1}'^f$. Then*:

$$pp\big(R_1'^f, \ldots, R_m'^f, R_{m+1}'^f\big) = pp\big(R_1'^f, \ldots, R_m'^f\big). \tag{29}$$

Proof. In negotiation round k when the buyer agent takes action critique, according to the protocol (see line 7 in Fig. 2) and Definition 12, the cut level is $pp(R_1'^f, \ldots, R_m'^f)$. Thus, by Definition 5, $\mu_{R_{m+1}'^f}(v_{var(R_{m+1}'^f)}) \geqslant pp(R_1'^f, \ldots, R_{m+1}'^f)$. And by Lemma 3, we have:

$$\frac{\rho(R_{m+1}'^f)}{\rho_{\max}} \diamond \mu_{R_{m+1}'^f}(v_{var(R_{m+1}'^f)}) \geqslant \mu_{R_{m+1}'^f}(v_{var(R_{m+1}'^f)}) \geqslant pp\big(R_1'^f, \ldots, R_{k'}'^f\big).$$

Thus, by Definition 12,

$$\begin{aligned} & pp\big(R_1'^f, \ldots, R_m'^f, R_{m+1}'^f\big) \\ &= \min\left\{pp\big(R_1'^f, \ldots, R_m'^f\big), \frac{\rho(R_{m+1}'^f)}{\rho_{\max}} \diamond \mu_{R_{m+1}'^f}(v_{var(R_{m+1}'^f)})\right\} \\ &= pp\big(R_1'^f, \ldots, R_m'^f\big). \qquad \square \end{aligned}$$

In the first round of negotiation, the buyer agent takes action select to generate a crisp constraint induced, at cut level 1, from the fuzzy constraint with the highest priority. Intuitively, if the seller finds a product satisfying the crisp constraint and offers it to the buyer, then in round 2 the buyer's potential payoff should be 1. The following theorem states that our definition for the buyer's potential payoff is consistent with this intuition.

Theorem 5. *Suppose in negotiation round* 1 *the crisp constraint that the buyer agent submits to the seller agent is induced from* R^f *at cut level* 1*. Then*

$$pp\big(R^f\big) = 1. \tag{30}$$

Proof. By Definition 5, $\mu_{R^f}(v_{var(R^f)}) \geqslant \sigma = 1$, while by Definition 4, $\mu_{R^f}(v_{var(R^f)}) \leqslant 1$. Therefore,

$$\mu_{R^f}(v_{var(R^f)}) = 1.$$

Thus, by Definitions 12 and 6,

$$pp\big(R^f\big) = \frac{\rho(R^f)}{\rho_{\max}} \diamond \mu_{R^f}(v_{var(R^f)}) = \frac{\rho(R^f)}{\rho(R^f)} \diamond 1 = 1 \diamond 1 = 1. \qquad \square$$

Finally, we give the calculation formula and property of the cut level used in action relax. Since we assume that all product attributes have finite possible values (see Definitions 2, 4, 6 and 9), all possible combinations of the attributes' values are finite. Thus, the satisfaction degrees associated with single attributes' values or their combinations take finite values in [0, 1]. Keeping this point in mind, we can formally define what it means to "reduce the cut level as little as possible" in action relax (see Definition 10):

Definition 17. Let the cut level in negotiation round k be $\sigma^{(k)}$. Suppose that in negotiation round $k+1$ the buyer agent relax fuzzy constraint R^f from crisp constraint R^c (which is induced from R^f at cut level $\sigma^{(k)}$) to crisp constraint R'^c. Let all the different satisfaction degrees of R^f constitute a finite series:

$$0 \leqslant \mu_1 < \cdots < \mu_m \leqslant 1. \tag{31}$$

Suppose $\mu_s < \sigma^{(k)} \leqslant \mu_{s+1}$. Then the cut level (denoted as $\sigma^{(k+1)}$), at which R^f induces R'^c, is defined as

$$\sigma^{(k+1)} = \mu_s. \tag{32}$$

This definition states that when taking action relax the buyer agent minimises the decrease of the cut level at which the relaxed constraint is induced. This implies that the decrease of the buyer's potential payoff is minimised.[11] In fact, we have the following theorem:

Theorem 6. *In negotiation round k, let the buyer's potential payoff be $pp^{(k)}(R_1'^f, \ldots, R_m'^f)$. Suppose in round $k+1$ the buyer agent takes action* relax *and submits to the seller agent a crisp constraint $R_m''^c$ which is induced from $R_m'^f$ at cut level $\sigma^{(k+1)}$ (given by* (32)*). In negotiation round $k+1$, let the buyer's potential payoff be $pp^{(k+1)}(R_1'^f, \ldots, R_m'^f)$. Then*

$$pp^{(k)}\big(R_1'^f, \ldots, R_m'^f\big) > pp^{(k+1)}\big(R_1'^f, \ldots, R_m'^f\big), \tag{33}$$

and the decrease is minimised.

Proof. Let $v_X^{(k+1)}$ and $v_X^{(k+2)}$ denote the attributes' values of the two products the seller agent offers in negotiation round $k+1$ and $k+2$, respectively. Thus, by Definitions 5 and 16, we have:

$$\mu_{R_m'^f}\big(v_{var(R_m'^f)}^{(k+1)}\big) > \mu_{R_m'^f}\big(v_{var(R_m'^f)}^{(k+2)}\big).$$

Then, by property (1) of priority operators listed in Definition 6, we have:

$$\frac{\rho(R_m'^f)}{\rho_{\max}} \diamond \mu_{R_m'^f}\big(v_{var(R_m'^f)}^{(k+1)}\big) > \frac{\rho(R_m'^f)}{\rho_{\max}} \diamond \mu_{R_m'^f}\big(v_{var(R_m'^f)}^{(k+2)}\big).$$

Thus, by Definition 12,

$$\begin{aligned}
&pp^{(k)}\big(R_1'^f, \ldots, R_m'^f\big) \\
&= \min\left\{pp\big(R_1'^f, \ldots, R_m'^f\big), \frac{\rho(R_m'^f)}{\rho_{\max}} \diamond \mu_{R_m'^f}\big(v_{var(R_{m+1}'^f)}^{(k+1)}\big)\right\} \\
&> \min\left\{pp\big(R_1'^f, \ldots, R_m'^f\big), \frac{\rho(R_m'^f)}{\rho_{\max}} \diamond \mu_{R_m'^f}\big(v_{var(R_{m+1}'^f)}^{(k+2)}\big)\right\} \\
&= pp^{(k+1)}\big(R_1'^f, \ldots, R_m'^f\big).
\end{aligned}$$

And by Definition 16, the decrease from $\mu_{R_m'^f}(v_{var(R_m'^f)}^{(k+1)})$ to $\mu_{R_m'^f}(v_{var(R_m'^f)}^{(k+2)})$ is minimised. So, the decrease from $pp^{(k)}(R_1'^f, \ldots, R_m'^f)$ to $pp^{(k+1)}(R_1'^f, \ldots, R_m'^f)$ is also minimised. □

[11] From Definition 17 we can clearly see that when a constraint is relaxed the procedure is basically one of hill-climbing. However, according to (31) it is clear that the hill is monotonic decreasing, and so it is impossible for the relaxing procedure to get stuck in local minima.

4.3. The negotiation outcome and information revelation

In this subsection, we analyse the properties of our negotiation algorithm and the solutions that it produces. In particular, we show that our negotiation system can find Pareto-optimal solutions and at the same time guarantee that the amount of private information revealed is minimised. We view Pareto-optimal solutions and minimal information revelation as important properties of our model because they accord closely with the main tenets of semi-competitive encounters as outline in Section 1. Notice that in the following all theorems are restricted to our agents' behaviour protocols and strategies.

In this context, a deal is a product that the seller agent offers to the buyer agent and the buyer agent accepts. According to standard business theory [4], the payoff that a seller gains from a deal is the profit of the product sold; and the payoff that a buyer gains is their satisfaction with the product bought. Formally, we have:

Definition 18. The *seller's payoff* is a sold product's profit (see Definition 9).

Definition 19. The *buyer's payoff* for a deal is the overall satisfaction degree of all constraints on the deal (as given by formula (7)).

Clearly, if an offered product is accepted in round $k+1$, the buyer's potential payoff (see Definition 12) in round k becomes its actually payoff for the deal.

Theorem 7. *During the negotiation encounter, the buyer's offers are generated in decreasing potential payoff order.*

Proof. According to the negotiation protocol given in Fig. 2, in the first round of a negotiation, the fuzzy constraint with the highest priority induces, at cut level 1, a crisp constraint that is submitted to the seller agent. That is, the buyer's potential payoff is 1 (by Theorem 5). Then, according to our negotiation protocol, in each of the subsequent rounds of the negotiation the buyer agent either submits a new crisp constraint or relaxes an already submitted one. According to Theorem 4, when submitting a new crisp constraint the buyer's potential payoff remains the same in the next negotiation round. According to Theorem 6, when it relaxes the constraint, the buyer's potential payoff is decreased as little as possible in the next negotiation round. So, the theorem holds. □

We now turn to the properties of the negotiation outcome.

Definition 20. A product p represents a Pareto-optimal solution if there is no other product p' such that at least one of the agent's payoffs is better for p' than it is for p and no agent's payoff is worse in p' than in p [50].

Theorem 8. *If there exists a solution* (*product*) *between seller and buyer agents then the protocol will terminate and a solution will be found through the protocol and strategies used by the participating agents. Moreover, the solution will be Pareto optimal.*

Proof. First, we prove that if a solution exists then the protocol must terminate and a solution must be found. Firstly, during the course of a negotiation, at the same level of the buyer's potential payoff, the buyer can only submit finite constraints since by Definitions 6 and 10 the buyer's constraints are finite and their domains are finite. Moreover, by Definition 9, the seller has a finite number of products and their associated rewards are also finite. Therefore, the searching procedure will not repeat infinitely at any given level of the buyer's potential payoff. Secondly, by Theorem 7, the buyer's offers are generated in decreasing potential payoff order and by Theorem 6 each decrease is minimised (i.e., between two potential payoffs there is no possible offer). Thirdly, according to the negotiation protocol of Fig. 1, at any point of a negotiation, if there is a product that can satisfy the buyer's submitted constraints, the seller must offer it to the buyer. Fourthly, according to the negotiation protocol given in Figs. 1 and 2, the search along the decreasing buyer's potential payoff order terminates when a solution that is accepted by the buyer agent is found. According to the first three points, all possible solutions are generated according to the negotiation protocol given in Fig. 2 and in order of the buyer's decreasing potential payoff. As a result, according to the fourth point, if the set of mutually acceptable solutions is not empty, a solution will be found by the search and the protocol will terminate.

Next, we prove that the solution found (the deal) is Pareto optimal. Assume that the deal is reached in the ith round. In other words, before the ith round no offers from the seller agent are acceptable to the buyer agent. Therefore, although before the ith round the possible solutions have higher payoffs for the buyer agent, these possible solutions are not acceptable deals. That is, no other solutions can give the buyer agent higher payoff. Moreover, according to the strategy of the seller's action generate used to produce the deal (see Definition 9), the deal that is reached in the ith round is the solution with the highest profit (payoff) for the seller among the solutions that can give the buyer the same payoff. That is, no other solutions can give the seller agent higher payoff. Therefore, the deal made through our protocol is Pareto optimal. □

This theorem states that given the buyer and seller's domain knowledge, our negotiation algorithm consisting of the negotiation protocol and strategies finds a Pareto-optimal solution.[12] We now turn to the information revealed during the encounter:

Theorem 9. *The buyer agent minimises the revelation of its requirement model* (*see Definition* 10) *during the negotiation.*

Proof. We have the following four points:

(1) According to the buyer's behaviour protocol (as shown in Fig. 2), in each round the buyer agent submits at most one constraint in its requirement model, and once it accepts the seller's offer, it does not submit any more constraints and ends the negotiation process. So, the buyer agent minimises the number of constraints that are revealed to the seller agent.

[12] The actual solution can be calculated only after the domain knowledge specifications of the seller and buyer agents (see Definitions 9 and 10) are actually instantiated. Therefore, it is not possible to determine the actual solution in this abstract setting.

(2) According to the way that action relax operates (see Definition 10), when the buyer agent relaxes a fuzzy constraint it minimises the degree of relaxation. If the degree of relaxation is not minimal, more compound labels (refer to Definitions 2 and 4) that satisfy the fuzzy constraint more or less will be revealed to the seller agent. Therefore, when relaxing a constraint, minimising the degree of relaxation implies that the revelation of the compound labels of the fuzzy constraint is also minimised.

(3) After the first round of negotiation, when submitting a constraint provided by action critique (see Definition 10), the buyer agent minimises the revelation of compound labels of the fuzzy constraint that induces the crisp constraint. In fact, if the cut level used to induce the crisp constraint from a fuzzy constraint is increased, although the potential payoff of the buyer agent does not decrease, the potential payoff of the seller agent may decrease. Thus, it is impossible to obtain a Pareto-optimal solution finally. This conflicts with Theorem 8.

(4) In the first round of negotiation, the buyer agent submits a crisp constraint provided by action select (see Definition 10). At that time, the cut level, which is used to induce the crisp constraint from a fuzzy constraint, is 1. So, at that moment the revelation of the compound labels of the fuzzy constraint is minimised.

In summary, during the course of a negotiation, the buyer agent minimises the number of constraints revealed. Moreover, for each fuzzy constraint that induces a submitted constraint, the buyer agent minimises the revelation of the compound labels of the fuzzy constraint. In addition, only fuzzy constraints in the buyer's requirement model are revealed to the seller agent. So, the theorem holds. □

Theorem 10. *The seller agent minimises the revelation of its product model* (*see Definition* 9) *during the negotiation.*

Proof. According to the seller's behaviour protocol (as shown in Fig. 1), in each round the seller agent reveals one product's attributes and attached restriction (refer to Definition 9) to the buyer agent. Moreover, when a product is revealed to the buyer agent for the first time, the reward associated with the product is not revealed. Additionally, the profit of a product is never revealed to the buyer agent. So, the theorem holds. □

When being taken together, Theorems 9 and 10 state that our negotiation model minimises the amount of private information revealed during the encounter. When combined with Theorem 8, we believe our system should be trusted by both sellers and buyers since it produces fair outcomes for them and minimises the disclosure of private information.

5. An accommodation renting scenario

To demonstrate the operation of our model and to show its practicality we have implemented a prototype systems for an accommodation renting scenario. Here a prospective tenant (student, the buyer) wants to rent some accommodation from a real estate agent (the seller). This scenario was chosen for two reasons. Firstly, it is readily comprehensible to most people. Secondly, such a scenario is a typical semi-competitive one. The seller agent tries to ensure the buyer agent rents accommodation which gives it the most profit, while the buyer agent tries to rent accommodation that most satisfies its constraints. That is, both agents try to obtain the best deal they can since they are both self-interested. To this end, they should minimise the revelation of their private information since it could stop them from getting good deals (see Section 5.3 for more discussion). However, since the seller needs to build or maintain his reputation (this should equate with more money in the long term) and the buyer needs to settle down as soon as possible (being homeless is not a good feeling), it is also necessary for them to cooperate to a certain extent in the negotiation.

5.1. The domain knowledge

We start by detailing the buyer's domain knowledge $\mathcal{KD}$ (see Definition 10) and the seller's domain knowledge $\mathcal{G}$ (see Definition 9) for this context.

First consider the buyer agent's domain knowledge $\mathcal{KD} = (\mathcal{C}, \mathcal{D}, \tau)$. Let us suppose the buyer agent wants to rent accommodation on behalf of its user who is a student. The buyer's requirement model $\mathcal{C}$ (refer to Definition 10) consists of two crisp constraints:[13]

R_1: "*distance* $\leqslant$ 15(minutes-walk)" with priority $\rho(R_1) = 0.37$,

R_2: "*rental-period* $\leqslant$ 12(months)" with priority $\rho(R_2) = 0.33$,

and fuzzy constraint R_3 on the combination of rental rate and period (shown in Table 1) which has priority $\rho(R_3) = 0.3$ and a relaxing threshold $\varrho(R_3) = 0.6$. Roughly speaking, the constraint expressed in Table 1 indicates that the more expensive the accommodation is, the shorter the acceptable rental period should be. So, this constraint actually captures the student's preference order on the trade-offs between rental rate and rental period. Thus, for example, the following are potential trade-offs for the agent: (1) the accommodation with rental rate £200 and rental period 12 months (line 1 in Table 1), (2) the accommodation with rental rate £265 and rental period 6 months (line 3 in Table 1), and (3) the accommodation with rental rate £325 and rental period 0.25 months (line 9 in Table 1). However, the buyer prefers the first option to the other two (because its satisfaction degree is bigger).

The buyer's profile model $\mathcal{B}$ (refer to Definition 10) consists of the following facts: the student is a female, she absolutely dislikes smoking, she likes a pet to the degree of 60%,[14]

[13] Notice that crisp constraints can be regarded as a special case of fuzzy constraints.

[14] The concept of liking a pet to a degree of 60% is clearly unnatural. When estimating uncertainties, people are often reluctant to provide real numbers chosen from a predefined range. Instead they feel more comfortable with a qualitative estimate [35,41] (i.e., a linguistic term). So, when answering the question how much they like a pet, people usually answer with one of the following linguistic terms: *very much, not much, a lot*, and so on. A linguistic term usually corresponds to a fuzzy set [68]. However, for simplicity, this paper treats a linguistic term as a number in $[0, 1]$. For example, "*very much*" may correspond to 60%, and thus "like a pet to a degree of 60%" means "like a pet very much". However, although this does not limit the generality of our model, it clearly limits its applicability since it is not easy for users to use real numbers to express the degree to which a proposition is true. This limitation could be removed by exploiting Zadeh's linguistic truth propositional logic [68]. However, in this case, the computation framework of linguistic fuzzy constraint satisfaction problems [25] would also need to be employed.

Table 1
Fuzzy constraint on rent and rental period

View Soft Constraints

Rate range (£)	Rental period (mo...	Satisfaction degree
0 < rate <= 250	12	100%
250 < rate <= 260	8	90%
260 < rate <= 270	6	80%
270 < rate <= 280	4	70%
280 < rate <= 290	3	60%
290 < rate <= 300	2	40%
300 < rate <= 310	1	30%
310 < rate <= 320	1/2	20%
320 < rate <= 330	1/4	10%
rate > 330	0	0%

Priority: 0.3 Relaxing threshold: 0.6 Return

Table 2
Accommodation information held by the seller agent

Accommodation information

Flat ID	Distance	Rental Rate	Rental Period	Reward	Restriction	Profit
f1	25	150	any	No Reward	Female	40
f2	15	240	any	No Reward	No pet	60
f3	15	300	>= 24	No Reward	None	100
f4	8	255	>= 12	New Furniture	No pet	60
f5	10	270	any	No Reward	Female & No smoking	50
f6	12	290	any	Air-conditioner	None	55
f7	12	270	any	Telephone	None	45
f8	30	130	any	No Reward	None	35

OK

and a telephone, some new furniture and an air conditioner are valuable important for her to the degrees of 70, 30 and 40%, respectively.

The buyer's acceptability threshold τ (refer to Definition 10) for accommodation is 0.7.

Now we turn to the seller. Table 2 shows the seller's product model (i.e., the information about the available accommodation prepared by the seller agent). Here the unit of rental rate and profit (the money that a real estate agent can make by renting out an accommodation) is *pound*, the unit of distance to the university is *minute walk*, and the unit of rental period is *month*.

5.2. The negotiation process

The negotiation proceeds as follows.

Round 1.

Buyer: Select the constraint with highest priority as follows:

$$distance \leqslant 15, \tag{34}$$

then present the seller agent with a message to ask it to `find` accommodation which satisfies this constraint.

Seller: Receive the buyer's offer message, and accordingly update its current constraint set (empty at the moment) with constraint (34). Thus its current constraint set becomes:

$$\{distance \leqslant 15\}. \tag{35}$$

Then, generate a solution, accommodation f_3, to the CSP (35). Further, try to propose the attached restriction for the solution. Here there is none. Finally, present the buyer agent with a message to ask it to `check` whether accommodation f_3 is acceptable according to the information "*rental-period* $\geqslant 24$", "*distance* $= 15$", and "*rental-rate* $= 300$". Notice that f_2, f_4, f_5, f_6 and f_7 also satisfy the constraint, but the profit of accommodation f_3 is the highest, and hence it is the one which is presented.

Round 2.

Buyer: Receive the seller's offer message, and accordingly verify accommodation f_3 against its current constraints in the order of their priorities. This shows that the constraint

$$rental\text{-}period \leqslant 12 \tag{36}$$

is violated. Thus, present the seller agent with a message to ask it to `find` an accommodation which can satisfy constraint (36) as well.

Seller: Receive the buyer's offer message, and first update its current constraint set (35) with constraint (36). Thus, its current constraint set becomes:

$$\{distance \leqslant 15, rental\text{-}period \leqslant 12\}. \tag{37}$$

Then, generate a solution, accommodation f_2, to the CSP (37). Further, propose the attached restriction for the solution: "no pet". Finally, present the buyer agent with a message to ask it to `check` whether accommodation f_2 is acceptable according to the information "no pet", "*distance* $= 15$", and "*rental-rate* $= 240$". Notice that f_4, f_5, f_6 and f_7 also satisfy the constraints in (37), but f_2 and f_4 have the highest profit (f_2 is therefore *randomly* chosen).

Round 3.

Buyer: Receive the seller's offer message, and verify accommodation f_2 against its current constraint set. No violations are found. Second, evaluate the buyer's acceptability for f_2. According to the buyer's requirement model, f_2 completely satisfies all its constraints, and so by (17) along with priority operator (22), f_2's requirement satisfiability is

$$\begin{aligned}\alpha &= \min\left\{(\mu_{R_1}-1)\frac{\rho(R_1)}{\rho_{\max}}+1,\ (\mu_{R_2}-1)\frac{\rho(R_2)}{\rho_{\max}}+1,\ (\mu_{R_3}-1)\frac{\rho(R_3)}{\rho_{\max}}+1\right\}\\ &= \min\left\{(1-1)\times\frac{0.37}{0.37}+1,\ (1-1)\times\frac{0.33}{0.37}+1,\ (1-1)\times\frac{0.3}{0.37}+1\right\}\\ &= 1.\end{aligned}$$

According to the buyer's profile model, the student likes a pet to the degree of 60%, and so f_2's restriction obedience $\beta = 1 - 60\% = 0.4$. Since there is no reward associated with f_2, in (16) f_2's reward value $\gamma = 0$. Thus, by (16) we have:

$$\begin{aligned}\mathit{acceptability}(f_2) &= \min\{\alpha,\beta\} \oplus_P \big((1-\tau)\gamma+\tau\big)\\ &= \min\{1, 0.4\} \oplus_P \big((1-0.7)\times 0+0.7\big)\\ &= 0.4 \oplus_P 0.7\\ &= \frac{(1-0.7)\times 0.4\times 0.7}{(1-0.7)\times 0.4\times 0.7+0.7\times(1-0.4)(1-0.7)}\\ &= 0.4.\end{aligned}$$

This value is less than the buyer's acceptability threshold 0.7, and so f_2 is not acceptable. Thus, present the seller agent with a message to ask it to `refind` an accommodation which satisfies all the constraints the buyer has submitted so far.

Seller: Receive the buyer's offer message, and generate an alternative solution, accommodation f_4, which has the same profit as accommodation f_2 (i.e., f_4 is a trade-off solution[15] of f_2). Further, propose the attached restriction for the solution: "no pet". Thus, present the buyer agent with a message to ask it to `check` whether accommodation f_4 is acceptable according to the information "no pet", "*rental-period* $\geqslant 12$", "*rental-rate* $= 255$" and "*distance* $= 8$". Notice that f_5, f_6, and f_7 also satisfy the constraint, but f_4 has the highest profit and so it is the one that is chosen.

Round 4.

Buyer: Receive the seller's offer message, and verify accommodation f_4 against its constraints. In this case, find that its constraint

$$\mathit{rental\text{-}rate} \leqslant 250 \wedge \mathit{rental\text{-}period} = 12 \tag{38}$$

is violated since the current cut level for fuzzy constraint R_3 is 1 according to formula (20). Then, present the buyer agent with a message to ask it to `find` an accommodation which can also satisfy constraint (38). Notice that crisp constraint (38) is induced from the buyer's fuzzy constraint, as shown in Table 1, at cut level 1 (see Definition 5).

Seller: Receive the buyer's offer message, and update its current constraint set (37) with constraint (38). Thus its current constraint set becomes:

$$\{\mathit{distance} \leqslant 15, \mathit{rental\text{-}period} \leqslant 12,\ \mathit{rental\text{-}rate} \leqslant 250 \wedge \mathit{rental\text{-}period} = 12\}. \tag{39}$$

[15] From Table 2, we can see that the profit of accommodation f_2 and f_4 is the same (£60) and so the seller is indifferent between them. However, the two potential solutions differ in their various attributes. In fact, compared with f_2, f_4's distance to the university is closer (hence better), but its rental rate is bigger (hence worse).

Then, try to generate a solution to the CSP (39), but fail. Thus, propose the reward for the last solution f_4: the seller agent can provide the student with some new furniture. Then, present the buyer agent with a message to ask it to `check` again whether accommodation f_4 is acceptable according to the revealed information plus the reward "new furniture".

Round 5.

Buyer: Receive the seller's offer message, and *check* whether the acceptability for f_4 is increased sufficiently by the reward. According to the buyer's requirement model, by (17) along with priority operator (22), f_4's requirement satisfiability is:

$$\begin{aligned}\alpha &= \min\left\{(\mu_{R_1}-1)\frac{\rho(R_1)}{\rho_{\max}}+1, (\mu_{R_2}-1)\frac{\rho(R_2)}{\rho_{\max}}+1, (\mu_{R_3}-1)\frac{\rho(R_3)}{\rho_{\max}}+1\right\}\\ &= \min\left\{(1-1)\times\frac{0.37}{0.37}+1, (1-1)\times\frac{0.33}{0.37}+1, (0.9-1)\times\frac{0.3}{0.37}+1\right\}\\ &= 0.92.\end{aligned}$$

According to the buyer's profile model, the student likes a pet to the degree of 60% (so f_4's restriction obedience $\beta = 1-60\% = 0.4$), and the reward "new furniture" associated with f_4 is valuable to the degree of 30% (so f_4's reward value $\gamma = 0.3$). Thus, by acceptability formula (16) along with uninorm operator (10), we have:

$$\begin{aligned}\mathit{acceptability}(f_4) &= \min\{\alpha,\beta\}\oplus_P \big((1-\tau)\gamma+\tau\big)\\ &= \min\{0.92, 0.4\}\oplus_P \big((1-0.7)\times 0.3+0.7\big)\\ &= 0.4\oplus_P 0.79\\ &= \frac{(1-0.7)\times 0.4\times 0.79}{(1-0.7)\times 0.4\times 0.79+0.7\times(1-0.4)(1-0.79)}\\ &= 0.518.\end{aligned}$$

This value is less than the buyer's acceptability threshold 0.7. So, even with the associated reward, f_4 is still not acceptable. Thus, present the seller agent with a message to ask it to `refind` an alternative which can satisfy all the buyer's constraints submitted so far.

Seller: Receive the buyer's offer message, and try to generate a new solution to the CSP (39), but fail. Moreover, the reward associated with last solution f_4 has been mentioned to the buyer agent. Thus, present the buyer agent with a message to ask it to `relax` some constraints. Notice only f_1, f_2 and f_8 can satisfy the constraint "*rental-rate* $\leqslant 250$". However, f_1 and f_8 cannot satisfy the constraint "*distance* $\leqslant 15$". f_2 can satisfy all constraints, but in the third round of the negotiation it was refused by the buyer agent because the buyer's acceptability for it was less than its threshold. So, the seller agent cannot generate a new solution.

Round 6.

Buyer: Receive the seller's offer message, and relax its fuzzy constraint (as shown in Table 1) from "*rental-rate* $\leqslant 250 \wedge$ *rental-period* $= 12$" to "*rental-rate* $\leqslant 260 \wedge$ (*rental-period* $= 12 \vee$ *rental-period* $= 8$" (i.e., first reduce the cut level of the fuzzy constraint from 1 to 0.9,[16] and then by formula (6) the relaxed result is obtained from Table 1). Then, present the seller agent with a message to ask it to `find` accommodation which can satisfy all the submitted constraints (including the newly relaxed one).

Seller: Receive the buyer's offer message, and update its current constraint set (39), by deleting constraint (38) and adding[17]

$$\textit{not}(\textit{rental-rate} \leqslant 250 \wedge \textit{rental-period} = 12), \tag{40}$$

$$\textit{rental-rate} \leqslant 260 \wedge (\textit{rental-period} = 12 \vee \textit{rental-period} = 8). \tag{41}$$

Thus its current constraint set becomes:

$$\begin{aligned} &\{\textit{distance} \leqslant 15, \textit{rental-period} \leqslant 12,\ \textit{not}(\textit{rental-rate} \leqslant 250 \wedge \textit{rental-period} = 12), \\ &\ \textit{rental-rate} \leqslant 260 \wedge (\textit{rental-period} = 12 \vee \textit{rental-period} = 8)\}. \end{aligned} \tag{42}$$

Then, try to generate a solution to the CSP (42), but fail again. Moreover, the reward associated with last solution f_4 has already been mentioned to the buyer agent. Thus, present the buyer agent with a message to ask it to further `relax` some constraints.

Round 7.

Buyer: Receive the seller's offer message, and relax the fuzzy constraint (as shown in Table 1) from "$\textit{rental-rate} \leqslant 260 \wedge (\textit{rental-period} = 12 \vee \textit{rental-period} = 8)$" to "$\textit{rental-rate} \leqslant 270 \wedge (\textit{rental-period} = 12 \vee \textit{rental-period} = 8 \vee \textit{rental-period} = 6)$" (i.e., first reduce the cut level of the fuzzy constraint from 0.9 to 0.8, and then by formula (6) the relaxed result is obtained from Table 1). Then, present the seller agent with a message to ask it to `find` accommodation according to the result of the relaxation.

Seller: Receive the buyer's offer message, and update its current constraint set (42) by deleting constraint (41) and adding constraints

$$\textit{not}\big(\textit{rental-rate} \leqslant 260 \wedge (\textit{rental-period} = 12 \vee \textit{rental-period} = 8)\big),$$

$$\textit{rental-rate} \leqslant 270 \wedge (\textit{rental-period} = 12 \vee \textit{rental-period} = 8 \vee \textit{rental-period} = 6).$$

Thus, its current constraint set (42) becomes:

$$\begin{aligned} &\{\textit{distance} \leqslant 15, \textit{rental-period} \leqslant 12, \\ &\ \textit{not}(\textit{rental-rate} \leqslant 250 \wedge \textit{rental-period} = 12), \\ &\ \textit{not}\big(\textit{rental-rate} \leqslant 260 \wedge (\textit{rental-period} = 12 \vee \textit{rental-period} = 8)\big), \\ &\ \textit{rental-rate} \leqslant 270 \wedge (\textit{rental-period} = 12 \vee \\ &\qquad \textit{rental-period} = 8 \vee \textit{rental-period} = 6)\}. \end{aligned} \tag{43}$$

[16] From Table 1, we can see that the second highest satisfaction degree of the fuzzy constraint is 0.9. So, according to the principle of reducing the cut level as little as possible when relaxing a constraint (see Definitions 10 and 16), we should reduce the cut level of the fuzzy constraint from 1 to 0.9 (instead of, for example, 0.95, or others).

[17] Since accommodation satisfying "$\textit{rental-rate} \leqslant 250 \wedge \textit{rental-period} = 12$" also satisfies "$\textit{rental-rate} \leqslant 260 \wedge (\textit{rental-period} = 12 \vee \textit{rental-period} = 8)$", "not($\textit{rental-rate} \leqslant 250 \wedge \textit{rental-period} = 12$)" must be added when the seller deletes "$\textit{rental-rate} \leqslant 250 \wedge \textit{rental-period} = 12$" and adds "$\textit{rental-rate} \leqslant 260 \wedge (\textit{rental-period} = 12 \vee \textit{rental-period} = 8)$".

Then, generate a solution, accommodation f_5, to the CSP (43), which has an attached restriction "female and no smoking". Finally, present the buyer agent with a message to ask it to check whether accommodation f_5 is acceptable according to the information "female and no smoking", "*rental-rate* = 270" and "*distance* = 10". Notice that from Table 2 we can see f_5 and f_7 both satisfy the constraints in (43), but f_5's profit (£55) is higher than f_7's (£45).

Round 8.

Buyer: Receive the seller's offer message, and *check* accommodation f_5. According to the buyer's requirement model, by (17) along with priority operator (22), f_5's requirement satisfiability is:

$$\begin{aligned}\alpha &= \min\left\{(\mu_{R_1}-1)\frac{\rho(R_1)}{\rho_{\max}}+1,\,(\mu_{R_2}-1)\frac{\rho(R_2)}{\rho_{\max}}+1,\,(\mu_{R_3}-1)\frac{\rho(R_3)}{\rho_{\max}}+1\right\}\\ &= \min\left\{(1-1)\times\frac{0.37}{0.37}+1,\,(1-1)\times\frac{0.33}{0.37}+1,\,(0.8-1)\times\frac{0.3}{0.37}+1\right\}\\ &= 0.838.\end{aligned}$$

According to the buyer's profile model, the student can completely obey the restriction condition "female and no smoking", and so f_5's restriction obedience $\beta = 1$. According to buyer's profile model, there is no reward associated with f_5, and $\gamma = 0$. Thus, by the acceptability formula (16) along with uninorm operator (10) we have:

$$\begin{aligned}\mathit{acceptability}(f_5) &= \min\{\alpha,\beta\}\oplus_P \big((1-\tau)\gamma+\tau\big)\\ &= \min\{0.838, 1\}\oplus_P \big((1-0.7)\times 0+0.7\big)\\ &= 0.838\oplus_P 0.7\\ &= \frac{(1-0.7)\times 0.838\times 0.7}{(1-0.7)\times 0.838\times 0.7+0.7\times(1-0.838)(1-0.7)}\\ &= 0.838. \end{aligned}\qquad (44)$$

This value is greater than the buyer's acceptability threshold 0.7. Thus, present the seller agent with a message to tell it that a deal has been made and that it can end its negotiation process.

Seller: Receive the buyer's offer message, and *end* the negotiation process.

To summarise this negotiation, the changes of data for the buyer and seller agents are given in Tables 3 and 4, respectively. In these tables, *rr*, *rp* and *d* stand for "rental rate", "rental period" and "distance", respectively.

5.3. Observations

After the negotiation exchange for this scenario has been viewed, there are two obvious questions to ask: (1) Why do we need to bother with this involved interchange? (2) Why

Table 3
The change of the buyer's data during the negotiation

	Received offer	Violated constraint	Acceptability	Cut level	Potential payoff	Offer constraint	Offer performative
Round 1				1		$+(d \leqslant 15)$	`find`
Round 2	f_3	$rp \leqslant 12$	0	1		$+(rp \leqslant 12)$	`find`
Round 3	f_2		0.4		1		`refind`
Round 4	f_4	$rr \leqslant 250 \wedge rp = 12$		1		$+(rr \leqslant 250 \wedge rp = 12)$	`find`
Round 5	f_4	$rr \leqslant 250 \wedge rp = 12$	0.518		0.92		`refind`
Round 6				0.9		$-(rr \leqslant 250 \wedge rp = 12)$ $+(\neg(rr \leqslant 250 \wedge rp = 12))$ $+(rr \leqslant 260 \wedge (rp = 12 \vee rp = 8))$	`find`
Round 7				0.8		$-(rr \leqslant 260 \wedge (rp = 12 \vee rp = 8))$ $+(\neg(rr \leqslant 260 \wedge (rp = 12 \vee rp = 8)))$ $+(rr \leqslant 270 \wedge (rp = 12 \vee rp = 8 \vee rp = 6))$	`find`
Round 8	f_5		0.838		0.838		`deal`

Table 4
The change of the seller's data during the negotiation

	Received constraint set	Candidate solutions	Offer solution	Potential payoff	Revealed information	Reward	Offer performative
Round 1	$d \leqslant 15$	$f_2, f_3, f_4, f_5, f_6, f_7$	f_3	100	$rp \geqslant 24, d = 15, rr = 300$	none	`check`
Round 2	$d \leqslant 15, rp \leqslant 12$	f_2, f_4, f_5, f_6, f_7	f_2	60	no pet, $d = 15, rr = 240$	none	`check`
Round 3	$d \leqslant 15, rp \leqslant 12$	f_4, f_5, f_6, f_7	f_4	60	no pet, $rp \geqslant 12, d = 8, rr = 255$	none	`check`
Round 4	$d \leqslant 15, rp \leqslant 12, rr \leqslant 250 \wedge rp = 12$	$\emptyset$	f_4	60	no pet, $rp \geqslant 12$ $d = 8, rr = 255$	new furniture	`check`
Round 5	$d \leqslant 15, rp \leqslant 12, rr \leqslant 250 \wedge rp = 12$	$\emptyset$					`relax`
Round 6	$d \leqslant 15, rp \leqslant 12, \neg(rr \leqslant 250 \wedge rp = 12), rr \leqslant 260 \wedge (rp = 12 \vee rp = 8)$	$\emptyset$					`relax`
Round 7	$d \leqslant 15, rp \leqslant 12, \neg(rr \leqslant 250 \wedge rp = 12), \neg(rr \leqslant 260 \wedge (rp = 12 \vee rp = 8)), rr \leqslant 270 \wedge (rp = 12 \vee rp = 8 \vee rp = 6)$	f_5, f_7	f_5	50	female, no smoking, $d = 10, rr = 270$	none	`check`

is negotiation necessary for this problem at all? In this subsection, we will discuss both questions.

From the above negotiation procedure, we know that f_1, f_2, f_3, f_4 and f_8 are not acceptable to the buyer agent, but f_5 is. Actually, f_6 and f_7 are also acceptable to the buyer agent. In fact, according to the buyer's requirement model, by (17) along with priority operator (22), f_6's requirement satisfiability is:

$$\begin{aligned}\alpha &= \min\left\{(\mu_{R_1}-1)\frac{\rho(R_1)}{\rho_{\max}}+1,\ (\mu_{R_2}-1)\frac{\rho(R_2)}{\rho_{\max}}+1,\ (\mu_{R_3}-1)\frac{\rho(R_3)}{\rho_{\max}}+1\right\}\\ &= \min\left\{(1-1)\times\frac{0.37}{0.37}+1,\ (1-1)\times\frac{0.33}{0.37}+1,\ (0.6-1)\times\frac{0.3}{0.37}+1\right\}\\ &= 0.676.\end{aligned}$$

According to the buyer's profile model, f_6's restriction obedience and reward value are $\beta = 1$, and $\gamma = 0.7$, respectively. Thus by acceptability formula (16) along with uninorm operator (10) we have:

$$\begin{aligned}\mathit{acceptability}(f_6) &= \min\{\alpha,\beta\} \oplus_P \big((1-\tau)\gamma+\tau\big)\\ &= \min\{0.676, 1\} \oplus_P \big((1-0.7)\times 0.4+0.7\big)\\ &= 0.676 \oplus_P 0.82\\ &= \frac{(1-0.7)\times 0.676\times 0.82}{(1-0.7)\times 0.676\times 0.82+0.7\times(1-0.676)(1-0.82)}\\ &= 0.803. \end{aligned} \tag{45}$$

This value is greater than the buyer's acceptability threshold 0.7. So, f_6 can be accepted by the buyer agent.

Similarly, for f_7, according to the buyer's requirement model, by (17) along with priority operator (22), f_7's requirement satisfiability

$$\begin{aligned}\alpha &= \min\left\{(\mu_{R_1}-1)\frac{\rho(R_1)}{\rho_{\max}}+1,\ (\mu_{R_2}-1)\frac{\rho(R_2)}{\rho_{\max}}+1,\ (\mu_{R_3}-1)\frac{\rho(R_3)}{\rho_{\max}}+1\right\}\\ &= \min\left\{(1-1)\times\frac{0.37}{0.37}+1,\ (1-1)\times\frac{0.33}{0.37}+1,\ (0.8-1)\times\frac{0.3}{0.37}+1\right\}\\ &= 0.838.\end{aligned}$$

f_7's obedience and reward value are $\beta = 1$ and $\gamma = 0.7$, respectively. Thus by formula (16) along with uninorm operator (10), we have:

$$\mathit{acceptability}(f_7) = \min\{\alpha,\beta\} \oplus_P \big((1-\tau)\gamma+\tau\big) \tag{46}$$

$$\begin{aligned} &= \min\{0.838, 1\} \oplus_P \big((1-0.7)\times 0.7+0.7\big)\\ &= 0.838 \oplus_P 0.91\\ &= \frac{(1-0.7)\times 0.838\times 0.91}{(1-0.7)\times 0.838\times 0.91+0.7\times(1-0.838)(1-0.91)}\\ &= 0.957. \end{aligned} \tag{47}$$

Table 5
The buyer's and seller's evaluations for f_5, f_6 and f_7

	Buyer's acceptability	Seller's profile
f_5	0.838	50
f_6	0.803	55
f_7	0.957	45

So, the acceptability for f_7 is greater than the buyer's acceptability threshold 0.7 and so f_7 can also be accepted.

Drawing this together, the buyer's and seller's evaluations for f_5, f_6 and f_7 are given in Table 5. It can be seen that from the buyer's perspective the best solution is f_7, but from the seller's view this is the worst solution. From the seller's perspective the best solution is f_6, but from the buyer's viewpoint this is the worst solution. Thus it can be seen that f_5 (the one provided by our model) represents a compromise solution. This is consistent with the intuitions of the principled model of negotiation.

Against this background, and to demonstrate the rationale for our negotiation model, consider the following alternative scenarios. Suppose that the buyer agent reveals all its constraints to the seller agent, but the seller agent reveals nothing to the buyer agent. In this case, the seller agent will give the buyer agent its best solution, f_6 plus the associated reward *air conditioner*, because the seller agent is self-interested. Conversely, suppose that the seller agent reveals all its data to the buyer agent, but the buyer agent reveals nothing. In this case the buyer agent will get its best solution, f_7 plus the associated reward *telephone*, because the buyer agent is also self-interested. So, in these two cases, one agent wins and the other agent loses (i.e., it is not a fair deal).

Another possible way of solving this problem would be to give the data of both the seller and buyer agents to a neutral third party. Then, this agent could be designed to make the compromise solution. This is exactly the role and function of many arbitration bodies [45,46]. However, although arbitration is perhaps a more efficient method of tackling this problem, it has a number of disadvantages. First, in many cases it is difficult for sellers and buyers to trust a mediator to act fairly on their behalf. Secondly, the privacy issue makes it almost impossible for sellers and buyers to expose their individually sensitive data to a third party. Since usually human negotiators are unwilling to disclose private information, decentralised methods for searching for Pareto-optimal solutions in negotiation problems are necessary [18]. Thirdly, arbitration represents a centralised solution to the problem with all the concomitant disadvantages that accrue. Thus, in this context, a distributed solution with incremental information revelation is necessary.

6. Related work

A number of models of bilateral multi-issue negotiation have been developed to date. Some of them are based on constraints, and some are not. Considering those that are not first. In the multi-issue negotiation model of Matos and Sierra [39], offers and counter offers are generated by case-based and fuzzy logic based strategies. In the Bazaar system developed by Zeng and Sycara [70], a multi-issue negotiation is explicitly modelled as a sequential decision making model. In fact, a Bayesian network is used to update the knowledge and beliefs each agent has about the environment and other agents, and offers and counter offers between agents are generated based on Bayesian probabilities. While both these models are adequate for dealing with the inherent uncertainties of bilateral negotiation, they cannot take advantage of the benefits of a constraint based approach as outlined in Section 1. This is because the users' requirements on attributes of a product/service that fuzzy constraints can easily capture are not represented in their systems. Matos and Sierra's model uses previous knowledge and information of the environment state, from a case base, to change its negotiation behaviour, a set of fuzzy rules to determine the values of the parameters of the negotiation model, and an evolutionary approach to determine which negotiation strategy is more successful. However, the issue of the users' requirements on the desired outcome of negotiation is not addressed in their work. As to Zeng and Sycara's model, while Bayesian networks are good at capturing the uncertain causal links between random variables, they do not have the function for modelling the users' requirements on attributes of products/services that our model does.

Moving onto the constraint based models (see Table 6 for a summary and comparison with our model).

- Tête-à-Tête (T@T) [15] is a multi-issue negotiation system. During a T@T negotiation, constraints on product features and constraints on merchant features are used to influence the decision of what and whom to buy from. However, the system itself is merely semi-autonomous. In fact, after evaluating and ordering the offers received from the sellers, the buyer agent just presents them to its user for consideration. It then passes the user's critiques to the seller agent in order to try and extract better offers. In contrast, our agents are autonomous in that they negotiate with each other on behalf of both sellers and buyers. Thus, they actually make the contract decisions themselves.
- Faratin et al. [9] dealt with trade-offs among multi-dimensional attributes during bilateral negotiations. Their criteria evaluation functions on a product's single issue can be regarded as a fuzzy constraint on this issue. However, unlike the work in this paper, fuzzy constraints over the combination of multiple issues of products are not considered. Moreover, in our work a buyer's offer is represented as a number of constraints which correspond to a set of possible solutions. In contrast, in their work an offer is represented as a single point solution. Generally speaking, even a small set of constraints can correspond to a large set of possible solutions. Thus, approaches that perform negotiation over sets of possible solutions are more efficient than approaches that perform negotiation on single point solutions, one at a time.[18]
- Kowalczyk and Bui [27] modelled the multi-issue negotiation process as constraint-based reasoning, and later on they extended [26] their approach into one based on the computational framework of FCSPs. However, unlike our work, their approach

[18] According to human negotiation theory [3,11], performing negotiation on single point solutions (that are what the negotiating agent wants) can be viewed as *positional bargaining*, while revealing the constraints (that need to be met) to the negotiation partner can be viewed as *interest-based negotiation*. In human negotiations, usually interest-based negotiations are also better than positional bargaining (the detailed discussion about the reasons can be found in [3,11]).

Table 6
The comparison of our work with related work

	T@T	Faratin Sierra Jennings	Kowalczyk Bui	Barbuceanu Lo	Ours
Requirement model					
Crisp/fuzzy constraint	crisp	fuzzy	crisp fuzzy	crisp	fuzzy
Constraints over single/multiple issues	multiple	single	multiple	multiple	multiple
Priority	no	no	no	no	yes
Profile model					
Buyer's profile model	no	no	no	no	fuzzy truth propositions
Reward	no	no	no	no	fuzzy truth proposition
Restriction	no	no	no	no	fuzzy truth proposition
Behaviour model					
Optimal	unknown	no	no	yes	yes
Autonomous	semi-	full-	full-	full-	full-
Negotiation over single/multiple solutions	single	single	single	single	multiple
Argument	no	no	no	no	reward

performs negotiation on individual solutions, one at a time. Moreover, in contrast to our work, their approaches do not guarantee that an optimal solution is found.

- Although the constraint-based multi-issue negotiation model of Barbuceanu and Lo [2] can guarantee that an optimal solution is found, their model also performs negotiation over possible solutions one by one. In contrast, our negotiations are carried out over fuzzy constraints of multiple issues of products, which is more efficient than doing it over single solutions. Moreover, the method of Barbuceanu and Lo is based on crisp constraints that are less flexible than the fuzzy constraints of our model.

In addition, the idea of a *reward*, from argumentation/persuasion-based negotiation models [28,29,42], is not integrated into any of the above previous work.

In the area of e-commerce, Sun et al. [54] also try to bridge CSPs and negotiation problems. Based on CSPs, they develop a model for order selection and negotiation in supply chains. In their model, the strategies for generating offers and counter offers are modelled as constraints. However, their model cannot handle the buyer's requirements (for a product) that are modelled by a FCSP. Thus they cannot take FCSPs' advantages for negotiation that we mentioned in Section 1.

The framework of CSPs has also been used for production selection and for matchmaking between sellers and buyers. For example: Ryu [49] proposed a product selection mechanism based on a hierarchy of constraints over product attributes, and Freuder and Wallace [12] describe a paradigm for matchmaking based on the constraint acquisition and satisfaction model. Here the matchmaker provides potential suggestions based on partial knowledge, and gains further knowledge through the evaluation of the suggestions from other agents. However, neither of the models involve negotiation.

In [33], we also developed a negotiation model based on the framework of FCSPs. However, there are important differences between this model and the one developed here. Firstly, the model in [33] is designed specially for meeting scheduling problems, whereas the current one is more widely applicable. Secondly, in [33] each agent has a FCSP and they negotiate for a common optimal solution for each FCSP. In the current model, one agent has the set of domains of variables, and another agent has the set of fuzzy constraints. The two agents negotiate for a solution that maximises the sum of their payoffs. Thirdly, in [33] the negotiation is performed over single solutions, whereas in the current model the negotiation is performed over multiple solutions implied by constraints. The latter is more efficient. Fourthly, in [33] the negotiation protocol is more restricted than that in the current model (e.g., it has no concepts of counter offer and reward).

Generally speaking, the various constraint-based negotiation processes can be regarded as a type of Distributed Constraint Satisfaction Problem (DCSP) where the domain, variables and constraints are distributed among multiple agents. Yokoo and his colleagues [63–66] have designed various algorithms to search for solutions to DCSPs. However, all these algorithms are based on the assumption that agents are not at all self-interested and so they always communicate constraints and modify their local solutions cooperatively. However, in many applications, agents have a degree of self-interest and so may be motivated to not disclose all their constraints for selfish reasons (as shown in Section 5.3). The ensuing constraint-based negotiation models, therefore, implement another class of algorithms to search solutions to a DCSP/DFCSP because they operate under different assumptions, namely that the information exchanged between agents may be incomplete and uncertain.

Finally, according to the convention in human negotiation [53,57], our protocol is designed as a non-increasing payoff protocol. That is, during the course of a negotiation, the payoff (benefit) that one negotiation agent expects is non-increasing. This is similar to the idea of the monotonic concession protocol proposed in [47]. However, their protocol assumes that the negotiating agents know each others' utility preferences, and so during the course of a negotiation they are moving in the direction of non-increasing utilities that opponents expect from the negotiation. In contrast, in our framework, the negotiating agents do not have such complete knowledge, and during the course of an encounter they move in the direction of non-increasing payoff that they expect from the viewpoint of their own preferences.

7. Conclusions and future work

This paper developed a prioritised fuzzy constraint based model for bilateral, multi-issue negotiations in a semi-competitive environment. The roles and benefits of the fuzzy constraint based approach can be summarised as follows:

- Fuzzy constraints are a natural means of modelling the buyer's requirements over products' single issues and the combination of the products' multiple issues. They are also appropriate for modelling trade-offs between different issues of a product.
- During negotiation it is often the case that a seller's offer only partially satisfies or violates the buyer's constraints. In other words, it is often unavoidable that the buyer has to relax its constraints. The computational framework of fuzzy constraints is ideally suited for capturing this process.

Compared with previous work, the model presented in this paper is novel in four aspects:

- It exploits the notion of prioritised fuzzy constraint satisfaction problems as the basic representation scheme. Previous models have used CSPs and FCSPs but not PFCSPs. The full power of the PFCSP model is needed to obtain all the benefits noted in Section 1.
- It enables negotiation to be carried out over fuzzy constraints of multiple issues of a product. This is more efficient than negotiation that is carried out over single point solutions.
- It guarantees that the outcome of the negotiation is Pareto optimal, yet the participating agents reveal minimal information about their preferences and constraints.
- It incorporates the concept of a *reward*, from argumentation/persuasion-based models. Rewards can be used to increase the buyer's acceptability for a product and thus increase the profit of the seller.

There are, however, a number of issues that require further investigation. Firstly, we would like to investigate the effect of endowing the buyer and/or the seller agents with alternative negotiation strategies. When submitting constraints, for example, the strategy of our buyer agent is to submit the highest priority one; when relaxing constraints, its strategy is to relax the lowest priority one. These strategies could be replaced by others that would make the agent more or less competitive/cooperative. Similarly, when choosing a product, our seller agent selects the one with the highest profit among those that completely satisfy the constraints submitted so far. This strategy might be changed, for example, to the similarity based mechanism of [9]. By using this strategy, the seller agent might be able to come to an agreement more quickly since it can focus its search in the acceptable region of the search space.

Secondly, since multi-criteria decision making (MCDM) [13,14] is a broadly similar framework to CSP based decision making, we would like to explore the possibility of whether the acceptance/counter-offer decision problem in negotiation can be reformulated as a decision problem that incorporates the advantages of both methods. For example, the idea that a constraint is on multiple attributes could be incorporated into MCDMs, and conversely some important concepts in MCDM, such as *veto* (if the evaluation on a certain criterion is high, it has no effect on the global evaluation, but if it is low, the global evaluation will be low too, whatever the evaluation of the other criteria [38]) and *favour* (if the evaluation on a certain criterion is low, it has no effect on the global evaluation, but if it is high, the global evaluation will be high too, whatever the evaluation of the other criteria [38]), may be introduced into PFCSPs. Such an integrated framework may enable us to finesse decision making in that some user's requirements (constraints on negotiation issues) become very sensitive when evaluating an opponent's offer during the course of a negotiation.

Thirdly, since our negotiating agents are knowledge intensive systems, it is natural that in the next step we will exploit techniques developed in the knowledge acquisition community in order to accurately extract the domain knowledge. This step is essential if software agents are to faithfully and appropriately represent their owners. However, to date, comparatively little attention has been paid to the problem of how users can impart sufficient knowledge into their agents such that they will be able to negotiate competently on their behalf. This is a key bottleneck that needs to be overcome if negotiating agents are to be widely deployed (see [31] for preliminary work in this direction).

Finally, we have adopted the CommonKADS Knowledge Engineering framework to help specify, organise and describe the knowledge intensive components of our agents. We regard this as more than an expository convenience and are working toward the construction of a library of such components that will provide us with a compositional methodology to support the specification, development and implementation of software agents.

Acknowledgements

The authors would like to thank the anonymous referees for their insightful comments that have significantly improved the quality of the paper. The work is partially supported by Hewlett Packard Research Labs and by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (RGC Ref. No. CUHK4304/98E).

References

[1] R. Bapna, P. Goes, A. Gupta, Insights and analyses of online auctions, Comm. ACM 44 (11) (2001) 42–50.
[2] M. Barbuceanu, W.-K. Lo, A multi-attribute utility theoretic negotiation architecture for electronic commerce, in: Proceedings of the Fourth International Conference on Autonomous Agents, 2000, pp. 239–246.
[3] S.P. Cohen, Negotiating Skills for Managers, McGraw-Hill, New York, 2002.
[4] P.H. Collin, Dictionary of Business, 3rd Edition, Peter Collin Publishing, 2001.

[5] R. Das, J. Hanson, J. Kephart, G. Tesauro, Agent-human interactions in the continuous double auction, in: B. Nebel (Ed.), Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01), Vol. 2, Seattle, WA, 2001, pp. 1169–1176.
[6] D. Dubois, H. Fargier, H. Prade, Propagation and satisfaction of flexible constraints, in: R. Yager, L. Zadeh (Eds.), Fuzzy Sets, Neural Networks and Soft Computing, Van Nostrand Reinhold, New York, 1994, pp. 166–187.
[7] D. Dubois, H. Fargier, H. Prade, Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty, Appl. Intelligence 6 (4) (1996) 287–309.
[8] D. Dubois, H. Prade, Qualitative possibility theory and its applications to constraint satisfaction and decision under uncertainty, Internat. J. Intelligent Syst. 14 (1999) 45–61.
[9] P. Faratin, C. Sierra, N.R. Jennings, Using similarity criteria to make issue tradeoffs in automated negotiations, Artificial Intelligence 142 (2) (2002) 205–237.
[10] T. Finin, R. Fritzson, D. McKay, R. McEntire, KQML as an agent communication language, in: Proceedings of the Third International Conference on Information and Knowledge Management, 1994, pp. 456–463.
[11] R. Fisher, W. Ury, Getting to Yes: Negotiating an Agreement without Giving in, Random House Business Books, 1981.
[12] E.C. Freuder, R.J. Wallace, Constraint-based strategies for matchmakers, in: Proc. AAAI-98 Workshop on Recommender Systems, Madison, WI, 1998, pp. 47–50.
[13] R. Fuller, C. Carlsson, Fuzzy multiple criteria decision making: Recent developments, Fuzzy Sets and Systems 78 (2) (1996) 139–153.
[14] M. Grabisch, Fuzzy integral in multicriteria decision making, Fuzzy Sets and Systems 69 (3) (1995) 279–298.
[15] R.H. Guttman, P. Maes, Agent-mediated integrative negotiation for retail electronic commerce, in: P. Noriega, C. Sierra (Eds.), Agent Mediated Electronic Commerce, in: Lecture Notes in Artificial Intelligence, Vol. 1571, Springer, Berlin, 1998, pp. 70–90.
[16] R.H. Guttman, P. Maes, Cooperative vs. competitive multi-agent negotiation in retail electronic commerce, in: M. Klusch, G. Weiß (Eds.), Cooperative Information Agents II, in: Lecture Notes in Artificial Intelligence, Vol. 1435, Springer, Berlin, 1998, pp. 135–147.
[17] M. He, H.F. Leung, N.R. Jennings, A fuzzy logic based bidding strategy in continuous double auctions, IEEE Trans. Knowledge Data Engrg. (2003), to appear.
[18] P. Heiskanen, H. Ehtamo, R.P. Hämäläinen, Constraint proposal method for computing Pareto solutions in multi-party negotiations, European J. Oper. Res. 133 (1) (2001) 44–61.
[19] J.M. Hiltrop, S. Udall, The Essence of Negotiation, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[20] N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, C. Sierra, M. Wooldridge, Automated negotiation: Prospects, methods and challenges, Internat. J. Group Decision Negotiation 10 (2) (2001) 199–215.
[21] N.R. Jennings, K. Sycara, M. Wooldridge, A roadmap of agent research and development, Autonomous Agent and Multi-Agent Systems 1 (1998) 7–38.
[22] R. Keeney, H. Raiffa, Decision with Multiple Objectives: Preferences and Value Tradeoffs, Wiley, New York, 1976.
[23] E.P. Klement, R. Mesiar, E. Pap, On the relationship of associative compensatory operators to triangular norms and conorms, Internat. J. Uncertainty Fuzziness Knowledge-Based Syst. 4 (2) (1996) 129–144.
[24] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[25] R. Kowalczyk, On linguistic fuzzy constraint satisfaction problems, in: L.A. Zadeh, J. Kacprzyk (Eds.), Computing with Words in Intelligent Information Systems, Physica-Verlag, 1999, pp. 305–321.
[26] R. Kowalczyk, On negotiation as a distributed fuzzy constraint satisfaction problem, in: Proceedings of the Third International Symposium on Soft Computing for Industry of the World Automation Congress, 2000, pp. 631–637.
[27] R. Kowalczyk, V. Bui, On constraint-based reasoning in e-negotiation agents, in: F. Dignum, U. Cortés (Eds.), Agent-Mediated E-Commerce III, in: Lecture Notes in Artificial Intelligence, Vol. 2003, Springer, Berlin, 2000, pp. 31–46.
[28] S. Kraus, Strategic Negotiation in Multiagent Environments, MIT Press, Cambridge, MA, 2001.
[29] S. Kraus, K. Sycara, A. Evenchik, Reaching agreements through argumentation: A logical model and implementation, Artificial Intelligence 104 (1–2) (1998) 1–69.

[30] Y.-M. Li, Z.-K. Shi, Remarks on uninorm aggregation operators, Fuzzy Sets Syst. 114 (2000) 377–380.

[31] X. Luo, N.R. Jennings, N. Shadbolt, Acquiring tradeoff preferences for automated negotiations: A case study, Submitted for publication.

[32] X. Luo, H.-F. Leung, J.H.-M. Lee, A new axiomatic framework for prioritized fuzzy constraint satisfaction problems, in: R. Mizoguchi, J. Slaney (Eds.), Topics in Artificial Intelligence: Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence, in: Lecture Notes in Artificial Intelligence, Vol. 1886, Springer, Berlin, 2000, p. 795.

[33] X. Luo, H.-F. Leung, J.H.-M. Lee, Theory and properties of a selfish protocol for multi-agent meeting scheduling using fuzzy constraints, in: Proceedings of the 14th European Conference on Artificial Intelligence, Humboldt University, Berlin, Germany, 2000, pp. 373–377.

[34] X. Luo, H.-F. Leung, J.H.-M. Lee, N.R. Jennings, Prioritised fuzzy constraint satisfaction problems: Axioms, instantiation and validation, Fuzzy Sets Syst. 136 (2) (2003) 151–188.

[35] X. Luo, C. Zhang, N.R. Jennings, A hybrid model for sharing information between fuzzy, uncertain and default reasoning models in multi-agent systems, Internat. J. Uncertainty Fuzziness Knowledge-Based Syst. 10 (4) (2002) 401–450.

[36] A.K. Mackworth, Consistency in networks of relations, Artificial Intelligence 8 (1) (1977) 99–118.

[37] P. Maes, R.H. Guttman, A.G. Moukas, Agents that buy and sell: Shoppers and sellers alike dispatch them into the digital bazaar to autonomously represent their best interests, Comm. ACM 42 (3) (1999) 81–91.

[38] J.-L. Marichal, On sugeno integral as an aggregation function, Fuzzy Sets and Systems 114 (3) (2000) 347–365.

[39] N. Matos, C. Sierra, Evolutionary computing and negotiating agents, in: Agent-Mediated Electronic Commerce, in: Lecture Notes in Artificial Intelligence, Vol. 1571, Springer, Berlin, 1998, pp. 126–150.

[40] J. Von Neumann, O. Morgenstern, Theory of Games and Economic Behaviour, Princeton University Press, Princeton, NJ, 1944.

[41] S. Parsons, Qualitative Methods for Reasoning under Uncertainty, MIT Press, Cambridge, MA, 2001.

[42] S. Parsons, C. Sierra, N.R. Jennings, Agents that reason and negotiate by arguing, J. Logic Comput. 8 (3) (1998) 261–292.

[43] S. Paurobally, J. Cunningham, Specifying the processes and states of negotiation, in: F. Dignum, C. Sierra (Eds.), Agent Mediated Electronic Commerce, in: Lecture Notes in Artificial Intelligence, Vol. 1991, Springer, Berlin, 2001, pp. 61–77.

[44] J. Peasall, The New Oxford English Dictionary, Oxford University Press, Oxford, 2001.

[45] D. Pruitt, Negotiation Behavior, Academic Press, New York, 1981.

[46] H. Raiffa, The Art and Science of Negotiation, Harvard University Press, Cambridge, MA, 1982.

[47] J.S. Rosenschein, G. Zlotkin, Rules of Encounter: Designing Conventions for Automated Negotiation among Computers, MIT Press, Cambridge, MA, 1994.

[48] A. Rubinstein, Perfect equilibrium in a bargaining model, Econometrica 50 (1) (1982) 97–109.

[49] Y.U. Ryu, A hierarchical constraint satisfaction approach to product selection for electronic shopping support, IEEE Trans. Syst. Man Cybernet. Part A: Systems and Humans 29 (6) (1999) 525–532.

[50] T. Sandholm, Distributed rational decision making, in: G. Weiss (Ed.), Multiagent Systems, MIT Press, Cambridge, MA, 1999, pp. 201–258.

[51] T.W. Sandholm, eMediator: A next generation electronic commerce server, in: Proceedings of the Fourth International Conference on Autonomous Agents, 2000, pp. 341–348.

[52] G. Schreiber, H. Akkermans, A. Anjewierden, R.D. Hoog, N. Shadbolt, W. van de Velde, B. Wielinga, Knowledge Engineering and Management: The CommonKADS Methodology, MIT Press, New York, 2000.

[53] P.T. Steel, T. Beasor, Business Negotiation: A Practical Workbook, Gower Publishing Limited, 1999.

[54] R. Sun, B. Chu, R. Wilhelm, J. Yao, A CSP-based model for integrated supply chain, in: Proceedings of AAAI-99 Workshop on Artificial Intelligence for Electronic Commerce, Orlando, FL, 1999, pp. 92–100.

[55] E.P.K. Tsang, Foundations of Constraint Satisfaction, Academic Press, New York, 1993.

[56] I. Unt, Negotiation without a Loser, Copenhagen Business School, 1999.

[57] R.J. Volkema, The Negotiation Toolkit: How to Get Exactly What You Want in Any Business or Personal Situation, Amacom, 1999.

[58] W. Walsh, M. Wellman, A market protocol for decentralized task allocation, in: Proceedings of the Third International Conference on Multi-Agent Systems, 1998, pp. 325–332.

[59] S. Willmott, M. Calisti, B. Faltings, S. Macho-Gonzales, O. Belakdhar, M. Torrens, CCL: Expressions of choice in agent communication, in: Proceedings of the Fourth International Conference on Multi-Agent Systems, 2000, pp. 325–332.
[60] M. Wooldridge, N.R. Jennings, Intelligent agents: Theory and practice, Knowledge Engrg. Rev. 10 (2) (1995) 115–152.
[61] R.R. Yager, Misrepresentations and challenges: A response to Elkan, IEEE Expert 9 (4) (August 1994) 41–42.
[62] R.R. Yager, A. Rybalov, Uninorm aggregation operators, Fuzzy Sets Syst. 80 (1996) 111–120.
[63] M. Yokoo, Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-Agent Systems, Springer, Berlin, 2000.
[64] M. Yokoo, E.H. Durfee, T. Ishida, K. Kuwabara, The distributed constraint satisfaction problem: Formalization and algorithms, IEEE Trans. Knowledge Data Engrg. 10 (5) (1998) 673–685.
[65] M. Yokoo, K. Hirayama, Distributed constraint satisfaction algorithm for complex local problems, in: Proceedings of the Third International Conference on Multi-Agent Systems, 1998, pp. 372–379.
[66] M. Yokoo, T. Ishida, Search algorithms for agents, in: G. Weiss (Ed.), Multiagent Systems, MIT Press, Cambridge, MA, 1999, pp. 165–201.
[67] L. Zadeh, Fuzzy sets, Inform. and Control 8 (1965) 338–353.
[68] L. Zadeh, The calculus of fuzzy restrictions, in: L.A. Zadeh, et al. (Eds.), Fuzzy Sets and Applications to Cognitive and Decision Making Processes, Academic Press, New York, 1975, pp. 1–39.
[69] D. Zeng, K. Sycara, How can an agent learn to negotiate?, in: J.P. Müller, M.J. Wooldridge, N.R. Jennings (Eds.), Intelligent Agents III: Agent Theories, Architectures, and Languages, in: Lecture Notes in Artificial Intelligence, Vol. 1193, Springer, Berlin, 1997, pp. 233–244.
[70] D. Zeng, K. Sycara, Bayesian learning in negotiation, Internat. J. Human-Computer Stud. 48 (1) (1998) 125–141.