

Heuristics for using CP-nets in utility-based negotiation without knowing utilities

Reyhan Aydoğ an · Tim Baarslag · Koen V. Hindriks ·
Catholijn M. Jonker · Pinar Yolum

Received: 21 February 2013 / Revised: 4 June 2014 / Accepted: 10 October 2014
© Springer-Verlag London 2014

Abstract CP-nets have proven to be an effective representation for capturing preferences. However, their use in automated negotiation is not straightforward because, typically, preferences in CP-nets are partially ordered and negotiating agents are required to compare any two outcomes based on a request and an offer in order to negotiate effectively. If agents know how to generate total orders from their CP-nets, they can make this comparison. This paper proposes heuristics that enable the use of CP-nets in utility-based negotiations by generating total orderings. To validate this approach, the paper compares the performance of CP-nets with our heuristics with the performance of UCP-nets that are equipped with complete preference orderings. Our results show that we can achieve comparable performance in terms of the outcome utility. More importantly, one of our proposed heuristics can achieve this performance with significantly smaller number of interactions compared to UCP-nets.

Keywords Automated negotiation · Qualitative preferences · CP-nets ·
Heuristic-based approaches

R. Aydoğ an (✉) · K. V. Hindriks · C. M. Jonker
Interactive Intelligence Group, Delft University of Technology,
Delft, The Netherlands
e-mail: R.Aydogan@tudelft.nl

T. Baarslag
Agents, Interaction and Complexity Group, University of Southampton, SO17 1BJ Southampton, UK
e-mail: T.Baarslag@soton.ac.uk

K. V. Hindriks
e-mail: K.V.Hindriks@tudelft.nl

C. M. Jonker
e-mail: C.M.jonker@tudelft.nl

P. Yolum
Department of Computer Engineering, Bogazici University,
Bebek, 34342 Istanbul, Turkey
e-mail: pinar.yolum@boun.edu.tr

1 Introduction

Conflict of interests occurs in many aspects of our life. *Negotiation*—the process of resolving conflicts [26, 32]—may take place in order to reach a consensus when conflict of interests exists among multiple stakeholders. Negotiating among participants may require multiple interactions especially when there are multiple issues to be agreed on. Some people may think that negotiation is a complex and time-consuming process, and some may believe that they cannot negotiate well and it would be better if someone else negotiates on behalf of them. For this purpose, we can use automated negotiation tools in which software agents negotiate on behalf of their users. Those automated negotiation tools might be incorporated in a group recommendation system where multiple users may have conflicting preferences, and it is necessary to find a mutual agreement that can serve as a recommendation to increase the users' satisfaction.

We expect such software to represent its user, understand her preferences, reason and act on her behalf. These key behaviors that we expect from this software match them ideally with software agents. While representing and reasoning about the user's preferences, there are several issues to be taken into account. First, the space of all possible negotiation outcomes grows exponentially with the number of attributes and their possible values. It may be infeasible to ask a user to order or rank all outcomes when the outcome space is large. Second, the user may have difficulty in assessing her preferences in a quantitative way [7]. Representing someone's preferences with numerical values is a difficult task for a human. Third, it is difficult to find a mathematical model for representing some preferences such as conditional preferences in which there are preferential dependencies between attributes. The above leads us to the fact that it is more effective and intuitive to use a qualitative preference model [3, 7].

Moreover, real-life negotiation problems might involve interdependencies; thus, it is also important to be able to represent conditional preferences. For instance, in a holiday scenario, the user's preference on transportation may depend on her holiday destination. When going to a close by location she might prefer a car, and for a far destination a plane. However, if she goes to a close by city, she may prefer taking a bus. As far as the aforementioned properties are concerned, a qualitative preference model, such as CP-nets [7] would be a good choice. CP-nets do not only provide us a natural and intuitive way of expressing the users' preferences but also allow representation of conditional preferences. However, CP-nets usually induce a partial preference ordering on the outcome space. That is, there are some incomparable outcome pairs with respect to a given CP-net. However, a negotiating agent should be able to compare each outcome to make its decision effectively. So when using a CP-net representation, one additional challenge is to develop mechanisms for an agent to negotiate with only partial preferences being available.

The main contribution of this paper is to develop methods to enable an agent to use qualitative partial preferences in negotiation. Usually, an agent negotiates according to a negotiation strategy, which dictates how offers are generated and which counter-offers are acceptable. To do this, a negotiating agent needs to evaluate outcomes with respect to a metric and act accordingly on this metric. Almost all of the existing negotiation strategies [5, 14, 16, 23, 27] work with utility functions that capture preferences in quantitative way. Basically, utility functions are used to assess the outcomes by mapping an outcome to a utility value, denoting to what extent the outcome is desired or useful for the user. Indeed, it is easy to compare outcomes with respect to their utility. Therefore, our solution for negotiating using a CP-net is to develop heuristics that map partial preference orderings of outcomes to the numeric assessment of outcomes in terms of estimated utilities. The estimated utilities

can be used by any negotiation approach based on utilities unless the approach uses the structure of the utility function explicitly. Therefore, our approach allows us to combine the best of two ideas, i.e., using CP-nets to capture the user's preferences and using the wealth of high-performance negotiation strategies that originally were developed for utility functions. From this perspective, the challenge has become to generate a suitable total ordering from a given partial ordering.

For our experiments, we employ the GENIUS (General Environment for Negotiation with Intelligent multi-purpose Usage Simulation) negotiation framework [22,30]. GENIUS provides an environment to evaluate and design negotiation strategies. We have extended GENIUS with CP-nets and added our heuristics to map CP-nets to total orderings. To obtain the qualitative preferences from the users, we added an elicitation tool for acyclic CP-nets.

Here, an important point is to be able to evaluate the performance of these heuristics fairly. To do this, we would need both a qualitative representation (which is a partial ordering) and a quantitative representation (or a total ordering) of a user's preferences. On the one hand, we can ask the user for her quantitative preferences and compare each negotiation outcome reached by the heuristics with respect to these quantitative preferences. On the other hand, one needs to ensure that the ordering of outcomes is consistent with the partial ordering induced from the CP-net. For comparison, we need to find a convenient quantitative model that is able to satisfy the CP-net relations and allows us to easily verify whether the total ordering provided by this representation is consistent with the ordering induced from a given CP-net. Fortunately, UCP-nets [6] have the aforementioned properties. Thus, UCP-nets can serve as ground truth in our setting. Domshlak et al. [13] state that UCP-nets might be used as the quantitative approximation of CP-nets. This claim also supports our choice of using UCP-nets as ground truth.

In our experiments, we used two ways of getting both the partial ordering and the total ordering representations of the preferences of a user. We generated these representations automatically (see Sect. 4.5), and we asked humans to construct their preferences in both a CP-net and in a UCP-net for a holiday (see Sect. 4.4). Thus, we could compare two cases:

1. The performance of agents when they apply heuristics on their users' qualitative preferences as CP-nets and negotiate with estimated utilities;
2. The performance of agents when they have access to their users' total preference orderings as UCP-nets.

In our experiments, two agents negotiate using the preference profiles we obtained. When the agent starts negotiating with the opponent agent, its negotiation strategy uses the estimated utilities to generate its offers and make a decision accordingly. As is customary, each agent knows only its own preferences and acts with respect to these preferences. In other words, the agents only consider their own preferences while negotiating and have no idea about the opponent's preferences. In our experiments, with the aim of comparing the performance of CP-nets (with a heuristic) with the performance of UCP-nets (the user's total ordering), the first agent (*Agent A*) has either a CP-net or a UCP-net, and negotiates with the second agent (*Agent B*) having a linear additive utility function.

We compare the performance by measuring the average overall utility gained by *Agent A*, the average number of interactions to reach a consensus and their worst-case counter parts. Our results show that most of our heuristics lead to negotiations that are almost as good as negotiations done with UCP-nets. In some cases, our heuristics can finish a negotiation earlier than UCP-nets. Interestingly, even the worst-case performances of the heuristics mostly lead to successful negotiations.

We can summarize our contributions as threefold: (1) We have developed a number of heuristics to be used with CP-nets in automated negotiation. Consequently, the agents can negotiate with partial preferences. (2) We have set up a test bed where the performance of the agents negotiating with partial preferences by applying a proposed heuristic can be compared fairly with the performance of the agents negotiating with the complete preferences. To achieve this, we have extended GENIUS framework. (3) Our experimental results showed that the agents applying the proposed heuristics on their partial preferences could negotiate almost as well as the agents having complete preferences. Furthermore, those results give insights in the tradeoff between using qualitative preferences in terms of CP-nets with the proposed heuristics and using qualitative preferences in terms of UCP-nets in negotiation.

The rest of this paper is organized as follows: Sect. 2 gives an introduction on CP-nets and UCP-nets. Section 3 explains the heuristics that we propose to use with CP-nets. Section 4 explains our experimental setup, metrics, and results. Finally, Sect. 5 discusses our work, and Sect. 6 concludes the paper with directions to future work.

2 Technical background

In this section, we give a brief introduction to two preference models, namely conditional preference networks (CP-nets) and UCP-Networks (UCP-nets).

2.1 Conditional preference networks (CP-nets)

CP-nets are graphical models for representing qualitative preferences in a compact way [7]. In CP-nets, each node represents an attribute (variable) and each edge denotes preferential dependency between nodes under the *ceteris paribus*, “all else being equal” assumption.

Here, if there is an edge from X to Y , X is called “parent node” and Y is called “child node”. The preference for child nodes depends on their parent nodes’ values. To express conditional preferences, each node is associated with a conditional preference table (CPT), which represents a total order on possible values of that node with respect to its parents’ values.

Consider our holiday domain in Example 1 and the CP-net depicted in Fig. 1. According to this CP-net, the user’s preference on location depends on season. CPT for *Location* shows that the user prefers a holiday in France over a holiday either in Turkey or in the Netherlands, and she prefers a holiday in Turkey over a holiday in the Netherlands in the winter season. However, when it is summer, she prefers a holiday in Turkey over either France or the Netherlands and a holiday in the Netherlands is preferred over that in France. Note that in CP-nets, each preference statement is interpreted under the “everything else being equal” assumption. The statement, “*Summer* is preferred over *Winter* for season”, means that if all other attributes such as location, hotel location, room type, duration, transportation are the same, an holiday in *Summer* is preferred over a holiday in *Winter*.

Example 1 For simplicity, our holiday domain has six attributes with a limited set of possible values per attribute. These are *Location*, *Hotel Location*, *Room Type*, *Season*, *Duration* and *Transportation*. *Location* denotes to the countries that Alice and Bob are considering for their holiday: **France**, **the Netherlands** and **Turkey**. *Hotel Location* can be near to **Sea** or **Historical Place** or **Mountain**. *Room Type* is categorized as **Non-smoking** and **Smoking**. For *Season* there are two values: **Summer** and **Winter**. The values for *Transportation* are **Car** and **Plane**. There are three possible values for *Duration*: **One week**, **Two weeks** and **Three weeks**.

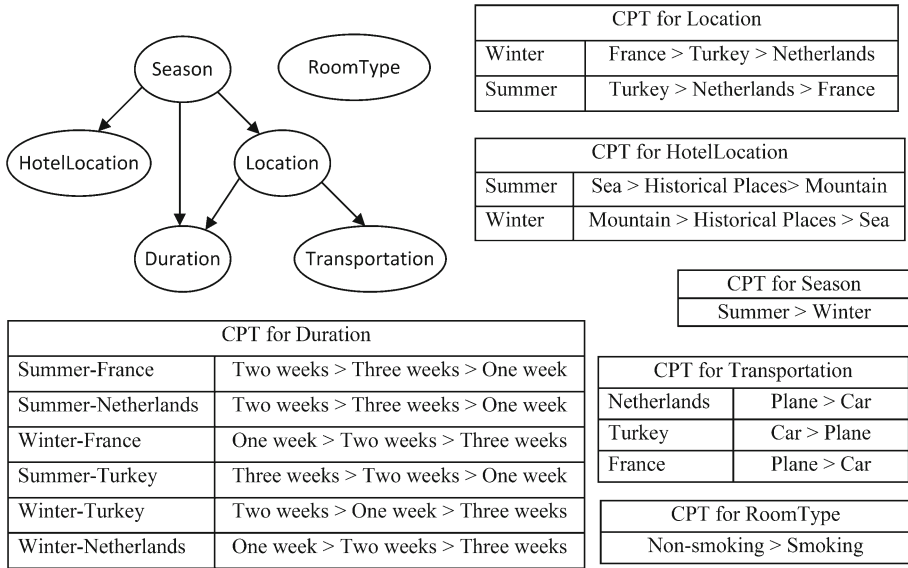


Fig. 1 A sample CP-net for our holiday domain

In acyclic CP-nets, there is only one best (optimal) outcome so it is straightforward to determine the best outcome (“outcome optimization query”). From ancestors to descendants, the most desired value for each attribute is chosen in order to get the best outcome. However, we need to check whether there exists an *improving flip* sequence from one outcome to another (and vice versa) to answer dominance queries (whether an outcome would be preferred over another). An improving flip is changing the value of a single attribute with a more desired value using CPT for that attribute. To illustrate this, consider the CP-net in Fig. 1. According to this CP-net, *Summer* is preferred over *Winter* for the season. Thus, we can apply an improving flip from $\langle \textit{Turkey, Sea, Non-smoking, Winter, One Week, Plane} \rangle$ to $\langle \textit{Turkey, Sea, Non-smoking, Summer, One Week, Plane} \rangle$ by changing the value of the season. If we cannot reach one outcome from another and vice versa via improving flip sequences, we cannot compare these two outcomes. For instance, we cannot compare $\langle \textit{Turkey, Sea, Non-smoking, Summer, Three Weeks, Plane} \rangle$ and $\langle \textit{Turkey, Sea, Non-smoking, Summer, One Week, Car} \rangle$ according to the CP-net in Fig. 1 since there is not any possible improving flip sequence between them. The fact that we may not be able to compare some outcomes is one of the challenges of using CP-nets in negotiation.

2.2 UCP-networks (UCP-nets)

Boutilier et al. propose a graphical preference model, namely UCP-net [6] by combining CP-nets with generalized additive independence models (GAI-models). Hence, UCP-nets are able to represent preferences in a quantitative way rather than representing simply preference ordering. GAI-models [18] perform dominance queries (whether an outcome would be preferred over another) straightforwardly, whereas CP-nets perform outcome optimization queries (maximal outcome) straightforwardly.

Figure 2 shows an example UCP-net that is consistent with CP-net seen in Fig. 1. Similar to CP-nets, we specify preferential dependency among attributes. Instead of specifying a total preference ordering over the values of each attribute according to their parents’ values

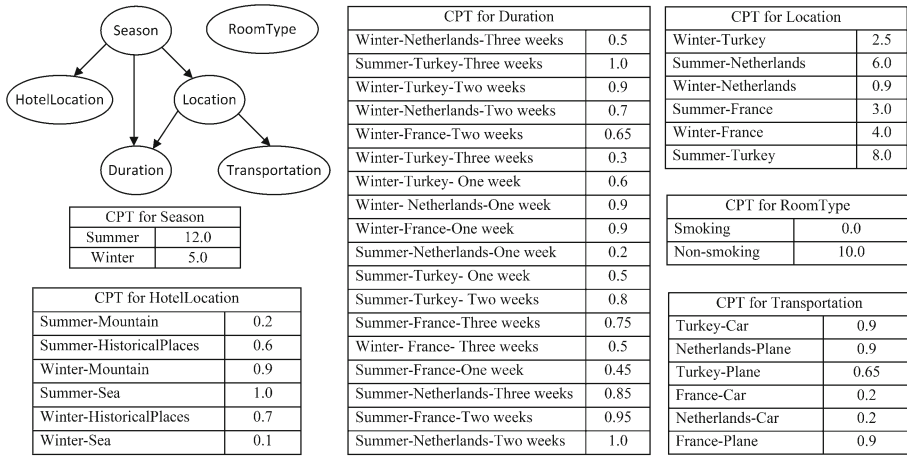


Fig. 2 A sample UCP-net

(conditions), we assign a real value (utility) for all values of each attribute by taking conditions into account. For instance, when season is *Winter*, the utility of *Turkey* as a location is specified as 2.5. Notice that the real values are not restricted to be between zero and one in this model.

The utility function $u(X_1, X_2, \dots, X_n)$ is represented in Eq. 1 where X_i is the i th attribute of the outcome, U_i denotes a set of parents of X_i and $f_i(X_i, U_i)$ represents a factor. Note that each different factor is treated as generalized additive independent of other factors. For example, our sample UCP-Net involves six factors f_1 (Season), f_2 (Location, Season), f_3 (Hotel Location, Season), f_4 (Room Type), f_5 (Duration, Season, Location) and f_6 (Transportation, Location). The utility of an outcome is estimated as the sum of these factors. For example, the utility of (*Winter, France, Mountain, No-smoking, Two weeks, Plane*) is equal to 21.45 ($=5.0 + 4.0 + 0.9 + 10.0 + 0.65 + 0.9$).

$$u(X_1, X_2, \dots, X_n) = \sum_i f_i(X_i, U_i) \tag{1}$$

In CP-nets, it is implicitly induced that an ancestor has higher priority over its descendants. Note that this property constitutes a key role in UCP-nets in which each attribute should dominate its children. When we assign utilities, we need to ensure that each node dominates its children. There are several ways of verifying whether the given network is a valid UCP-net (whether it satisfies the CP-relations among attributes).

One of the methods for verifying UCP-nets is to compute the values of *Maxspan* and *Minspan* for attributes and check whether Eq. 2 is satisfied or not. In this equation, X and Y are attributes and Y_i are the children of X . Note that $Maxspan(Y)$ is the maximum difference between the utilities of each possible values of Y for the given parent value. We can define *Minspan*(X) analogously.

$$Minspan(X) \geq \sum_i Maxspan(Y_i). \tag{2}$$

If the condition in Eq. 2 is satisfied for each attribute in the given network, we can say that it is a valid UCP-net. For example, in our sample UCP-net *Location* is a parent of both *Duration* and *Transportation*. For estimating the value of $Maxspan(Transportation)$, we evaluate the maximum difference between the utilities of the values of *Transportation* for each possible value of its parent [France: 0.7 ($=0.9-0.2$), Netherlands: 0.7 ($=0.9-0.2$) and Turkey: 0.25 ($=0.9-0.65$)] and choose the maximum among them ($Maxspan = 0.7$). Similarly we get the $Maxspan(Duration)$ as 0.8. To be able to get a valid UCP-net satisfying the CP-relationships, the difference between the utilities of each possible value of *Location* should be at least 1.5 ($=0.7+0.8$). In our example, it is equal to 1.6 ($=2.5-0.9$), which is higher than 1.5. We check this condition for each attribute. If the given UCP-net satisfies this condition for each attribute (CP-relationships among attributes), we say that the given UCP-net is a valid UCP-net.

Since we use normalized utility values between zero and one in our negotiation setting, we divide the utility of the outcome by the utility of the best outcome (whose utility is the highest). For this case, the utility of the best outcome $\langle Summer, Turkey, Sea, Non-smoking, Three weeks, Car \rangle$ is 32.9 ($=12.0+8.0+1.0+10.0+1.0+0.9$). Thus, the normalized utility of $\langle Winter, France, Mountain, No-smoking, Two weeks, Plane \rangle$ would be 0.65 ($=21.45/32.9$).

3 Proposed heuristics

Most of the negotiation strategies [5, 14, 16, 23, 27, 36] work with quantitative preferences such as *utility functions* proposed in the literature. However, it is desired for users to express their preferences qualitatively. Thus, we propose heuristics to use acyclic CP-nets (a qualitative preference model) in negotiation while agents still negotiate with their strategies using quantitative information, *utility* (a real value between zero and one). To do this, we generate estimated utilities from a given CP-net by applying our heuristics.

In our framework, a user's preferences are elicited as an acyclic CP-net and then a preference graph is induced from it. In this preference graph, each node denotes a possible outcome and each edge represents an improving flip. Note that a directed edge from one node to a second node shows that the second node is preferred over the first node. Therefore, the least desired (worst) outcome will be placed at the top of the preference graph (root node), whereas the optimal (best) outcome will be placed at the leaf node. Recall that there is only one best outcome in acyclic CP-nets. For intermediate nodes, we only compare the nodes having a path from others. The nodes having no path to each other cannot be compared under the "everything else being equal" assumption, CP-net semantics.

Consider the preference graph in Fig. 3 induced from the CP-net in Example 2. Here, the node $\langle Mountain, Summer, One week \rangle$ represents a one-week holiday in a hotel near a mountain during summer. It is the worst outcome according to the given preferences. Note that it is easy to find the worst outcome. We only need to choose the least desired values for each attribute by starting from parent attributes in the CP-net. While constructing the preference graph, we draw an edge from a node to another, if there is an improving flip from the former to the latter. It is seen that there is an edge from $\langle Mountain, Summer, One week \rangle$ to $\langle Sea, Summer, One week \rangle$. It can be inferred that a one-week holiday in a hotel near a sea in the summer is preferred over a one-week holiday in a hotel near a mountain in the summer. Recall that we cannot compare two outcomes if there are no paths (any sequences of improving flips) between them. For instance, $\langle Mountain, Summer, Two weeks \rangle$ and $\langle Mountain, Winter, Three weeks \rangle$ are not comparable according to the preference graph.

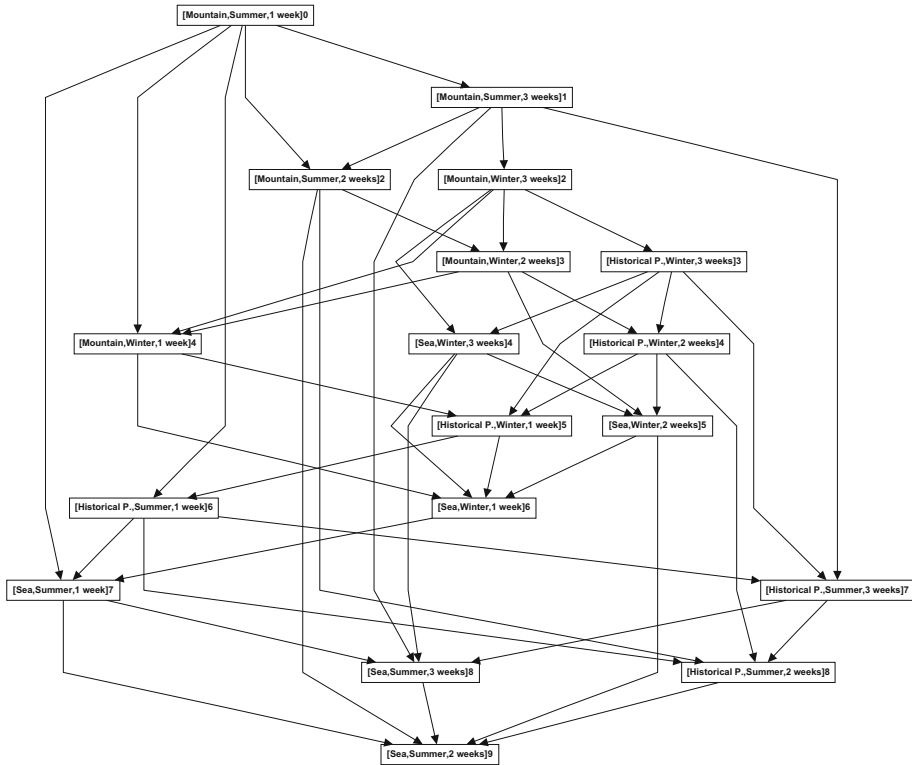


Fig. 3 Induced preference graph from the CP-net in Example 2

Example 2 For simplicity, assume we have only three attributes in the holiday domain: *Hotel Location*, *Season* and *Duration*. According to the user, her preferences on *duration* depend on *season* and her preferences on *season* depend on *hotel location* (hotel location \rightarrow season \rightarrow duration). For hotel location, she prefers a hotel near a sea over a hotel near historical places and a hotel near historical places is preferred over a hotel near a mountain ($Sea > Historical Places > Mountain$). If the hotel is near a mountain, she prefers a holiday in winter rather than in summer ($Mountain: Winter > Summer$). Otherwise, a holiday in summer is preferred over a holiday in winter ($[Sea \vee Historical Places]: Summer > Winter$). When it is winter, a one-week holiday is preferred over a two-week holiday and a two-week holiday is preferred over a three-week holiday ($Winter: One\ week > Two\ weeks > Three\ weeks$). When it is summer, a two-week holiday is preferred over a three-week holiday and a three-week holiday is preferred over a one-week holiday ($Summer: Two\ weeks > Three\ weeks > One\ week$).

It is worth to discuss the complexity of inducing a preference graph from an acyclic CP-net. The number of nodes grows exponentially with the number of attributes (issues). In fact, the total number of nodes is equal to $|Dom(x_1)| * |Dom(x_2)| * \dots * |Dom(x_n)|$ where we have n attributes and $|Dom(x_i)|$ denotes the number of possible values for the i th attribute. The total number of edges in the graph is equal to the number of all possible improving flips. For example, there will be $(|Dom(x_1)| - 1) + (|Dom(x_2)| - 1) + \dots + (|Dom(x_n)| - 1)$ outgoing edges of the root node (the least desired outcome). Note that the number of outgoing edges of the nodes are decreasing as the nodes become close to the leaf node and the number

of outgoing edges of the leaf node (optimal outcome) becomes zero. Since we start from the least desired outcome and continue with applying improving flips, it is straightforward to generate the preference graph.

Although inducing a preference graph from an acyclic CP-net seems to be costly and time-consuming, during the negotiation the agents will not be affected by this process because we construct the preference graph once while we elicit the user's preferences as a CP-net. Then the proposed heuristics will be applied to the induced preference graph in order to get the estimated utilities that will be used by the agent's strategy during the negotiation.

When the agent applies one of the proposed heuristics, each outcome in the outcome space is associated with an estimated utility, a real number between zero and one. Since those numerical values are comparable with each other, the agent will be able to compare all outcome pairs. For each outcome x, y , the agent interprets that the user prefers y over x if $EU(x) < EU(y)$ and that the user prefers x and y equally if $EU(x) = EU(y)$. Consequently, the agent will be able to employ a negotiation strategy requiring a total ordering of outcomes.

3.1 Depth heuristic (DH)

According to the induced preference graph from a given CP-net, we know that the root node keeps the worst choice and the leaf node keeps the best choice for the user. For each intermediate node, the node pointed by another node is more preferred over that node. Since we cannot compare the outcome pairs that do not have a path between them, we define a metric called *depth* to compare each outcome pair intuitively [1–3]. The depth of an outcome node indicates how far an outcome is from the worst choice. Here, our intuition is that “the more preferred an outcome is, the further it is from the worst outcome”. To measure the distance, we use each edge, improving flip, as a unit of distance and estimate the depth of an outcome node as the length of the longest path from the root node to that node. Thus, *depth* shows the highest distance from the worst choice node. The depth also shows the number of outcomes over which this one is preferred. For instance, if the depth of an outcome is three, we ensure that that outcome is preferred over at least three outcomes.

We have previously proposed several negotiation strategies based on capturing the *depth* of an outcome in the preference graph induced from a given CP-net [2]. In that study, we have concentrated on how an agent uses a CP-net in its negotiation strategy. For this purpose, we have first induced a preference graph from the given CP-net and applied several search methods such as *depth first search*, *depth limited search* in order to generate an offer and to decide whether a counter offer is acceptable or not. Notice that in that study using depth plays a key role in the proposed negotiation strategies. However, in this study, we use the concept of *depth* to produce estimated utilities of the outcomes regardless of the negotiation strategy. That is, the agent using this heuristic is able to apply any utility-based negotiation strategy.

The intuition here is that if there is an edge from x to y , we know that y is preferred over x and the depth of y is higher than that of x . According to this approach, the higher the depth of an outcome, the more likely it is preferred by the user. Further, if two outcomes are at the same depth, it is assumed that these outcomes are equally preferred by the user. We apply Eq. 3 to estimate the utility values between zero and one [1]. In short, the depth of a given outcome is divided by the depth of the preference graph (the highest depth) to obtain estimated utility of that outcome. For example, if we have a preference graph with a depth of 9 in Fig. 3, an outcome whose depth is equal to 4 will have utility of $0.44(=4/9)$ according to this approach.

$$U(x) = \frac{\text{Depth}(x, PG)}{\text{Depth}(PG)} \quad (3)$$

Since we update the depth of each node while inducing the preference graph, it is straightforward to estimate the utility of an outcome with respect to this heuristic. In other words, we can estimate the utilities in linear time on the basis of the induced graph.

3.2 Depth level heuristic (DLH)

As an alternative to Depth Heuristic (DH), we can take the number of outcomes at each depth into account and distribute the utility functions according to the density of the outcomes at each depth. This method takes the depth of an outcome and outputs a utility value between zero and one. A candidate utility function is formalized in Eq. 4 where $U(i)$ denotes the utility of outcomes at depth i , S_i is the number of outcomes at depth i and N is the number of all possible outcomes. We define the utility of the outcome at depth zero ($U(0)$) is equal to $1/N$.

$$U(i) = U(i - 1) + \frac{1}{N} \cdot S_i \quad (4)$$

It is worth noting that the Depth Level Heuristic gives the same ordering as the Depth Heuristic because $U(i) > U(i - 1)$ but the magnitude of the estimated utilities are different. This may lead to different negotiation results even if the agent applies the same negotiation strategy. For example, if the negotiation strategy only generates offers whose utility is higher than a predefined threshold value (reservation point), the agent applying DLH may generate different offers than the agent applying DH. Thus, it can be proven that the stratification of knowledge has a significant impact on the performance of some negotiation strategies. However, this is not true for some negotiation strategies, which consider the ordering of outcomes rather than to their extent. Consider a simple concession-based strategy in which the agent starts with the outcome having the highest utility and concedes over time (the second highest utility, the third highest utility and so on) up to reaching a consensus. Under these circumstances, the negotiation result would be the same for both heuristics, DH and DLH.

The complexity of DLH seems to be higher than that of DH since it involves the estimation of the utilities for the previous depths. In fact, the utility of an outcome is equal to the ratio of the number of outcomes whose depth is lower than or equal to the depth of the current node to the number of all possible outcomes ($U(i) = \frac{1+S_1+\dots+S_i}{N}$). Since we update the number of the outcomes at each depth while inducing the preference graph, we can estimate the utilities in linear time on the basis of the induced graph. Therefore, the complexity of DH and DLH would be equal.

Definition 1 For a given acyclic CP-net, a utility function U mapping outcomes to utilities in $[0, 1]$ is said to be *consistent* with the CP-net, if for every two outcomes a, b it holds that $U(b) > U(a)$ whenever $b > a$.

We show that estimated utilities satisfy the existing CP-net relations. That is, if an outcome b is preferred over another a with respect to a given CP-net, the estimated utility of b should be higher than that of a when we apply the Depth Heuristic. As a result, the ordering of the estimated utilities by these heuristic will be consistent with CP-net semantics.

Theorem 1 Let an acyclic CP-net C be given. The utility functions induced by the heuristics DH and DLH are consistent with C .

Proof Suppose that according to the CP-net \mathcal{C} with preference graph PG and root z , it holds that $b > a$. We need to show that $Utility(b) > Utility(a)$, for the utility functions $Utility$ induced by the heuristics. We will state the proof for the DH heuristic. The proof for DLH is analogous as these heuristics guarantee that $Utility(y) > Utility(x)$ whenever $Depth(y) > Depth(x)$.

As $b > a$, there exists a sequence $P_{a \rightarrow b}$ of improving flips $a < p_1 < \dots < p_k < b$, of nonzero length. We call such a sequence a path from outcome a to b . Furthermore, let $P_{z \rightarrow a}$ be a longest path from z to a . By definition, it holds that $Length(P_{z \rightarrow a}) = Depth(a)$.

Now let $P_{z \rightarrow b}$ be defined as the above path $P_{z \rightarrow a}$ from z to a , followed by the path $P_{a \rightarrow b}$ from a to b . We define $X_{z \rightarrow b}$ as the set of all paths from the root to b . Note that $P_{z \rightarrow b} \in X_{z \rightarrow b}$, and hence:

$$\begin{aligned} Depth(b) &= \max_{p \in X_{z \rightarrow b}} \{Length(p)\} \geq Length(P_{z \rightarrow b}) \\ &= Length(P_{z \rightarrow a}) + Length(P_{a \rightarrow b}) > Depth(a). \end{aligned}$$

Therefore, if $b > a$ then

$$Utility(b) = Depth(b)/Depth(PG) > Depth(a)/Depth(PG) = Utility(a).$$

□

3.3 Taxonomic heuristic (TH)

The previous heuristics (DH and DLH) consider that an outcome at a higher depth gets higher utility. That is, if there are two outcomes and the depth of the first outcome is higher than the depth of the second outcome, the utility of first outcome will be higher than the utility of the second outcome with respect to those heuristics. They also assume that outcomes at the same depth are equally preferred by the user. In contrast, the TH does not make these assumptions. According to this approach, the utility of an outcome is determined randomly but the estimated utilities should be consistent with CP-net semantics. That is, if there is a sequence of improving flips from the first outcome to the second outcome, the utility of the second outcome should be higher than the utility of the first outcome. Since there exist a sequence of improving flips from any node to its descendants, the descendants are preferred over the ancestors in the preference graph. In short, we generate the utility of an outcome randomly in accordance with the previous statements.

Equation 5 shows how the utility of an outcome (o_i) is estimated randomly. The utility of the worst outcome (root node) is set to zero directly. For other outcomes, we consider the highest utility of their ancestors and generate a random real that is greater than zero and less than or equal to one. The utility of an outcome is estimated as the sum of this randomly generated utility ($randomUtility(0.0, 1.0)$) and the highest utility of the outcome's ancestors ($Max(U(Ancestors(o_i)))$). Note that we calculate the utility of outcomes at lower depth earlier because we need to check ancestors' utility while estimating descendants and the depth of ancestors is lower than the depth of descendant. Before using these estimated utilities in negotiation, we normalize them by dividing them by the highest utility.

$$U(o_i) = Max(U(Ancestors(o_i))) + randomUtility(0.0, 1.0) \tag{5}$$

Similar to DLH, the estimation of the utilities for this heuristic is incremental. We first estimate the utilities of the outcomes that are in the upper level of the preference graph. If there exist m ancestors of an outcome, we check the utilities of these m outcomes and get the maximum one. Since we have already estimated the utilities of ancestors while estimating

the utility of the current node, choosing the maximum and adding a random real to it can be performed in linear time.

3.4 PageRank heuristic (PRH)

PageRank is a heuristic to rank Web Pages, to determine the ranks of the nodes in a graph whose nodes denote Web pages. The underlying intuition is that a Web page is considered important if other important pages links to this page [8]. We apply the same intuition to our problem: An outcome is more preferred if it is pointed by other preferred outcomes in the preference graph—if there are improving flip sequences from other preferred outcomes to that node.

According to this approach, we first apply PageRank algorithm and get PageRank value for each outcome. To estimate the PageRank of the i th service node, we use the formula in Eq. 6 where K_i denotes the nodes pointing to the node i , $P(j)$ is the rank of the j th node, N_j are the neighbors of the j and d is the damping factor, which we set to 0.85 as in the original paper [8].

$$P(i) = d \sum_{j \in K_i} \frac{P(j)}{|N_j|} + (1 - d) \quad (6)$$

Initially, each service node's PageRank is equal to 1.0. Then, we start the calculation of ranks according to recursive formula in Eq. 6. When the difference between two calculations is small enough (when the algorithm converges), we reach the final ranks. Since the estimated values may exceed one, we divide each value by the maximum value in order to get normalized values between zero and one. Table 1 shows the estimated normalized PageRank values for the preference graph depicted in Fig. 3.

After our initial experimentation, we realized that the distance between the best outcome and the second best outcome is relatively high, making it difficult for a strategy to make a choice other than the best one. To illustrate this, consider the normalized PageRank values in Table 1. The normalized PageRank value of the optimal outcome is 1.0 where that of the second best outcome is equal to 0.463. For example, if the agent's strategy lets it only offer the outcomes whose utility is higher than 0.6, it will never offer or accept the second best outcome during the negotiation. This situation may lead to a failure to reach a consensus. Thus, we do not prefer to use those normalized values directly as utilities.

To deal with this problem, we apply a clustering algorithm over the estimated normalized PageRank values and make each outcome inside the same cluster have at most a certain distance with others in the cluster. For this purpose, we use a clustering algorithm called X-means [12], which takes a range of number of clusters (minimum and maximum number of clusters) and outputs the best number of clusters and the clustering information (which data belongs to which cluster). Then, we sort the normalized PageRank values and distribute the values according to their clusters.

Equation 10 represents how we transform the sorted normalized PageRank values into utilities between zero and one where $U(i)$ denotes the utility of i th outcome, $NPR(i)$ denotes the normalized PageRank value of i th outcome and $c(i)$ represents which cluster the PageRank of i th outcome belongs to. Firstly, we define k equal distance intervals for each cluster where k is the number of clusters that we have. To illustrate this, following our running example, consider that we have 18 outcomes whose normalized PageRank values are shown in Table 1. When we apply the clustering algorithm (X-means), we obtain four different clusters ($k = 4$). As a result, the interval for each cluster is equal to 0.25. We estimate the upper bound for each cluster by applying the formula in Eq. 7 where c_i denotes the cluster

Table 1 Outcomes and their estimated normalized ranks

Outcome [Hotel location, season, duration]	Estimated PageRank	Normalized PageRank	Final estimated utility
[Sea, summer, two weeks]	1.786	1.000	1.000
[Sea, summer, one week]	0.827	0.463	0.750
[Sea, summer, three weeks]	0.742	0.415	0.738
[Sea, winter, one week]	0.654	0.366	0.726
[Historical places, summer, two weeks]	0.518	0.290	0.707
[Historical places, winter, one week]	0.386	0.216	0.500
[Sea, winter, two weeks]	0.360	0.202	0.497
[Historical places, summer, one week]	0.340	0.190	0.494
[Historical places, summer, three weeks]	0.324	0.181	0.491
[Mountain, winter, one week]	0.286	0.160	0.486
[Historical places, winter, two weeks]	0.261	0.146	0.250
[Mountain, winter, two weeks]	0.250	0.140	0.249
[Sea, winter, three weeks]	0.230	0.129	0.246
[Mountain, summer, two weeks]	0.213	0.119	0.243
[Historical places, winter, three weeks]	0.190	0.106	0.240
[Mountain, winter, three weeks]	0.187	0.105	0.240
[Mountain, summer, three weeks]	0.175	0.100	0.239
[Mountain, summer, one week]	0.150	0.084	0.235

number. According to this formula, the upper bound for each cluster (1, 2, 3 and 4) is equal to 0.25, 0.50, 0.75 and 1.0, respectively. When we estimate the utility of each outcome whose normalized PageRank value is the highest in their cluster, we use the upper bounds as the utilities of those outcomes (Case-1 Eq. 10). For example, the outcome whose PageRank value is the highest in the second cluster will get a utility of 0.50.

$$upper\ Bound(c_i) = \frac{1}{k} \cdot c_i \tag{7}$$

$$interval(i) = \frac{1}{k \cdot n_i} \tag{8}$$

$$dist(i, i + 1) = \frac{NPR(i + 1) - NPR(i)}{k} \tag{9}$$

Since our aim is to assign utilities in a way that the distance between utilities of consecutive outcomes in the same cluster can be at most a certain distance, we define another interval within each cluster by taking the number of outcomes inside that cluster into account. This interval is defined in Eq. 8 where n_i denotes the number of outcomes in the i th cluster. According to our example, since we have four outcomes in the third cluster, this interval is equal to 0.0625 (=0.25/4). When we assign utilities to outcomes, we consider both the distance between their normalized PageRank values and the estimated interval value for the current cluster. We calculate the distance between their normalized PageRank values by applying the formula in Eq. 9 where we divide the distance between PageRank values by the number of clusters. If the estimated distance is greater than the interval value for that cluster, we take the interval value as the distance as specified in Eq. 10 ($min(interval(c(i), dist(i + 1, i)))$).

As a result, the utility of current outcome is estimated by subtracting this distance from the utility of previous outcome. For example, in the third cluster, we have four outcomes with normalized PageRank value of 0.463, 0.415, 0.366 and 0.290. Since 0.463 is the maximum value in this cluster, it gets a utility of 0.75, which is the upper bound of the third cluster. The distance between the normalized PageRank values, 0.463 and 0.415 is equal to 0.012 according to Eq. 9. Since it is less than the interval value for this cluster (0.0625), we take 0.012 as the distance. As a result, the utility of that outcome (having a normalized PageRank value of 0.415) will be equal to 0.738 (=0.75–0.012).

$$U(i) = \begin{cases} \text{upper Bound}(c(i)) & \text{(Case-1)} \\ U(i + 1) - \min(\text{interval}(c(i)), \text{dist}(i + 1, i)) & \text{(Case-2)} \end{cases} \quad (10)$$

The complexity of this heuristic involves both the complexity of PageRank and clustering. The complexity of PageRank is specified as $O(\log N)$ in [25]. The overall complexity would be polynomial since we apply clustering to the sorted PageRank values.

3.5 Preferred outcomes heuristic (POH)

In this approach, while we estimate the utility of an outcome, we take the outcomes that are preferred over that outcome into account. For this purpose, we use a scoring function that not only considers the number of outcomes preferred over the current outcome but also takes into account how much they are preferred. This scoring function is the weighted sum where each weight denotes the importance of an outcome that is preferred over the current outcome. However, we do not know how much each outcome is preferred (importance of an outcome) since we only have the qualitative preferences of the user. Thus, in this approach, the depth of an outcome is considered as the importance of that outcome: The higher the depth of an outcome, the more likely it is preferred by the user.

Our intuition for this heuristic is that an outcome is less desired if there are many highly desired outcomes that are preferred over that outcome. To illustrate this, consider the outcomes in Fig. 3, $\langle \text{Sea, Summer, One-week} \rangle$ and $\langle \text{Historical Place, Summer, Three weeks} \rangle$. Their depth is equal to seven. According to our depth heuristic, they seem to be equally preferred. However, there are only two outcomes that are preferred over the former, whereas there are three outcomes that are preferred over the latter one. If we have to choose one of them, we may be inclined to prefer the former because there are less outcomes that are better than it. In addition to the number of outcomes preferred over the outcome, how much they are preferred is also important.

The formula in Eq. 11 calculates this score for each outcome where $SR(x)$ denotes the score of the outcome x . In this formula, we sum the depth of each outcome y that is preferred over x .

$$SR(x) = \sum_{y>x} \text{depth}(y) \quad (11)$$

If we divide the estimated scores by the score of the worst outcome (whose score is the maximum), we obtain normalized scores. Since lower scores are more desired, we obtain the utility of an outcome by subtracting the normalized score from one. This utility function is formalized in Eq. 12 where x is the outcome whose utility is calculated and z is the worst outcome (root node in the preference graph).

$$U(x) = 1 - \frac{SR(x)}{SR(z)} \tag{12}$$

To estimate the utility of an outcome at depth of d , we need to traverse all outcomes that are preferred over that outcome. That is, we need to search the subtree whose root is that outcome and whose depth is equal to at most $(Depth(PG) - d)$ where $Depth(PG)$ denotes the depth of the whole preference graph. For this purpose, Breath First Search [35] is applied whose complexity is $O(b^d)$ where b denotes the branching factor and d denotes the depth of tree. Thus, the complexity of estimating the utility of an outcome at depth of d is $O(b^{Depth(PG)-d})$.

3.6 Average all heuristic (AAH)

This heuristic takes an average of the estimated utilities by *Depth Level Heuristic*, *Taxonomic Heuristic*, *PageRank Heuristic* and *Preferred Outcome Heuristic*. It assumes that each heuristic is equally important. Therefore, the final utilities are defined by taking the average of the estimated utilities by these heuristics.

4 Experiments

To evaluate the proposed heuristics, we extend GENIUS [22,30], which is a platform for bilateral negotiation—General Environment for Negotiation with Intelligent multi-purpose Usage Simulation. The aim of the environment is to facilitate the design and the evaluation of negotiation strategies. GENIUS provides a negotiation simulation environment in which a researcher can setup a single negotiation session or a tournament using various negotiation domains and preference profiles from a repository and choose strategies for the negotiating parties. Furthermore, it allows to create a negotiation domain and preference profiles of the negotiating parties.

Our extension enables an agent to elicit its user’s preferences as an acyclic CP-net and to use utilities estimated by a chosen heuristic while negotiating with an opponent agent. In this setting, the platform also requests the user’s total ordering of outcomes as a UCP-net and evaluates each negotiation outcome for that agent based on the given UCP-net. The aim of this is to evaluate how well we can negotiate with the estimated total order from a CP-net using heuristics. Ideally, the best we can do is to generate the total order already given by the user through a UCP-net. In this regard, the UCP-net serves as ground truth in our comparisons. After an agent negotiates using its CP-net, we evaluate its performance as if we knew the correct total ordering (UCP-net). Note that the given UCP-net is consistent with the given CP-net since both preferences belong to the same user.

4.1 Experimental settings

In order to compare the performance of the heuristics, we investigate three test cases depicted in Fig. 4. In each test case, two agents *Agent A* and *Agent B* negotiate with each other. We ensure that *Agent A* negotiates with the same *Agent B* (having the same preference profile and strategy).

In the first case, *Agent A* is a dummy agent that does not know its user’s preferences. Thus, it generates estimated utilities purely randomly and negotiates with these estimated utilities. In principle, this corresponds to the Zero Intelligence agent that randomly generates bids as in for example [20,24]. This agent serves as a baseline for our comparisons.

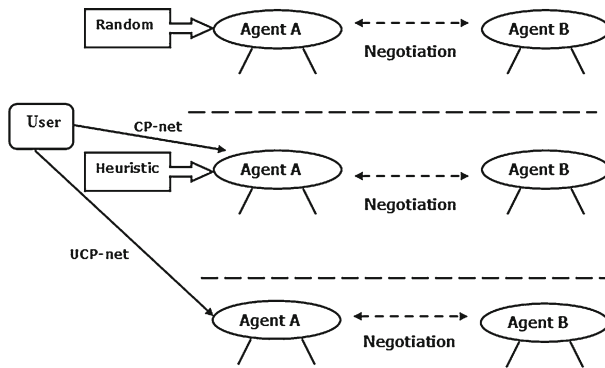


Fig. 4 Experiment setup for comparison of heuristics

In the second case, *Agent A* has a CP-net and applies one of the proposed heuristics (DH, DLH, TH, PRH, POH, AAH) to derive the estimated utilities. During the negotiation, the agent will act according to these estimated utilities. In both first and second cases, when the agents complete the negotiation, the negotiation outcome is estimated with respect to the real utilities represented via UCP-net. In the last case, *Agent A* has its user's real total preference orderings as a UCP-net (consistent with the CP-net and able to compare all outcomes). Thus, it uses the real utilities during the negotiation. Consequently, we are able to observe what the agent gets at the end of a negotiation when it applies heuristics on partial preference information (CP-net) versus when it has total preference information (UCP-net). In our experiments, each negotiation session ends in an agreement with respect to bids and all negotiation outcomes are evaluated according to the user's UCP-net since it serves as a ground truth. We use the UCP-net to assign utilities and thus are able to compare performance quantitatively.

4.2 Preference elicitation

Since the structure of the CP-nets (i.e., number of dependencies and hierarchical levels) may affect the performance of the heuristics, we use 10 different CP-nets and also 10 UCP-nets consistent with those CP-nets in our experiments. One of the significant factors affecting negotiation is the predictability of the opponent agent's preferences on some particular issues. For example, consider a buyer and a seller negotiating on a particular service. The seller may predict that the buyer prefers lower price over higher price and may act accordingly. We do not use the predictability in this study. Thus, none of the agents predict each other's preferences. We generate 10 different preference profiles for *Agent B*. Each *Agent A* negotiates with these 10 different *Agent Bs* individually. Because of the unpredictability of issues, it can be interpreted that we use 10 different domains having the same number of issues in our experiments.

For our experiments, we use our holiday domain explained in Example 1, which has 216 possible outcomes. To construct *Agent A*'s preference profiles (both CP-nets and UCP-nets), we elicited preferences of 10 people that are either students or faculty in Bogazici University or Delft University of Technology). We observe that people had difficulty in expressing their preferences as UCP-nets while they were more comfortable giving their preferences as CP-nets. That supports the fact that it is more effective and intuitive to use qualitative preference models from the users' point of view.

These 10 CP-nets are completely different from each other, different CP-net structures and different preference graphs (see Appendix). It is important to use different CP-nets in our experiments for the robustness of the results since the structure of the CP-nets (i.e., number of dependencies and hierarchical levels) may affect the performance of the heuristics.

In order to elicit the user's preferences as an acyclic CP-net, we need to enable the user to specify any preferential dependencies among attributes as well as to rank all possible attribute values with respect to their parents. To do this, we have developed an add-on tool to GENIUS that first asks the user to select the parents of an attribute from a given list. For instance, a user can express that her preferences on hotel location depends on the season. Here, season is the parent of the hotel location. Next, since we do not allow cyclic CP-nets, our tool checks whether the given dependencies are cyclic or not. If so, it enables the user to revise her choices. Finally, after eliciting acyclic preferential dependencies, the user selects the order of attribute values from a given list. After completing the elicitation of the CP-net, this tool generates a preference graph induced from the given CP-net, which is used by the proposed heuristic in order to generate estimated utilities.

Elicitation of the user's preferences as a UCP-net is performed in a similar way. In addition to specifying preferential dependencies, utility values are assigned to all possible attribute values with respect to their parents. This time, the tool verifies whether the given UCP-net is a valid UCP-net satisfying CP-relationships. To do this, it checks whether the condition for being a valid UCP-net (Sect. 2.2) is satisfied. If so, the elicitation of the user's preferences as a UCP-net is completed successfully. Otherwise, the system enables the user to revise the utility values. Thus, we ensure that all CP-net and UCP-net pairs used in our experiments are consistent.

Since *Agent B*'s preferences also affect the negotiation outcome significantly, we generate 10 different preference profiles (linear additive utility functions) randomly for *Agent B*. As a result, ten different *Agent As* negotiate with ten different *Agent Bs*. Note that each agent only knows its own preferences.

4.3 Negotiation settings

In principle, there are many different application domains and many different strategies for negotiation. Here we have chosen a holiday domain, since what is important is not the domain description itself but accommodating various intricate relations among attributes. Even though we have chosen a single domain, we provided variations on the preference structures by creating 10 different preference profiles that correspond to 10 different CP-nets.

We have selected a simple concession-based strategy for proposing bids during a negotiation session because we need to employ a simple strategy whose side effect is minimal on the evaluation results. That is, we need to ensure that the strategy itself does not heavily influence the outcome. In the mean time, the strategy should be sensitive to the order of the outcomes as much as possible so that we can detect how well we estimate the total order from a given partial order. Furthermore, it is also important to employ a negotiation strategy that is independent from the structure of the utility functions for a fair comparison. Thus, in our experiments, *Agent A* uses a simple concession-based strategy in which the agent starts with the outcome having the highest utility and concedes over time (the second highest utility, the third highest utility and so on) up to reaching an consensus. According to this strategy, *Agent A* will accept *Agent B*'s current counter offer if the utility of *Agent B*'s counter offer is higher than or equal to the utility of *Agent A*'s previous offer. Note that *Agent A* also remembers the best counter offer that is made by *Agent B*. While *Agent A* generates its own offer if the

utility of the current offer is lower than the utility of the *Agent B*'s best counter offer, *Agent A* requests *Agent B*'s best counter offer.

Agent B uses a simple random-based strategy because we can observe a variety of negotiation scenarios caused from the randomness. This is an important reason why we have 10 preference profiles for *Agent B*. By repeating the experiments using different preference profiles, we ensure that we report realistic performance results for *Agent A*. According to this strategy, the agent generates randomly an outcome whose utility is higher than a predefined threshold value. In our experiments, we choose the threshold value as 0.6 as a result of the analysis on *Agent B*'s utility distribution. During the negotiation, the threshold is updated when *Agent A* offers an outcome whose utility is higher than the current threshold. *Agent B* accepts *Agent A*'s counter offer when the utility of the counter offer is higher than or equal to the utility of *Agent B*'s previous offer. Note that this agent may produce the same offer several times because of randomness. This may lead to an increase in the number of rounds to reach a consensus. Thus, we determine a deadline for *Agent B*. If the number of rounds reaches 500, the agent accepts *Agent A*'s counter offer. In our experiments, very few negotiations (exactly 0.36 %) end due to this time out.

Furthermore, because of generation of offers randomly, we repeat each negotiation session 10 times and report the averages. It is worth noticing that using a random strategy we do not introduce a bias toward a particular strategy and therefore explore more of the strategy space. In addition, negotiating with the same *Agent B* 10 times gives the effect of negotiating with 10 *Agent Bs* having the same preferences but adapting different negotiation strategies. Consequently, we observe a variety of possible negotiations in our experiments.

4.4 Evaluation

We define four evaluation criteria for comparison:

- **Average utility of negotiation outcomes for *Agent A*:** Average utility is by far the most widely used performance measure in negotiation [4, 10, 21]. In our experiments, each *Agent A* negotiates with 10 *Agent Bs* having different preferences 10 times. As a result, for each *Agent A* there are 100 negotiations with *Agent B*. Note that during our experiments each negotiation session is completed successfully; that is, both agents have a consensus in all negotiations. According to this criterion, we evaluate each negotiation outcome (agreement) with respect to *Agent A*'s current preferences represented as a UCP-net so as to observe the performance of heuristics. Thus, the utility of each negotiation outcome is estimated over 100 negotiations and the average of these utilities is reported. Note that the utility of an outcome is a real value between zero and one.
- **Number of times that *Agent A* using a heuristic negotiates at least as well as *Agent A* having a UCP-net:** For each CP-net and UCP-net pair, we compare the negotiation outcomes for the agent using a heuristic with that for the agent having a UCP-Net. If the utility of a negotiation outcome for the agent using a heuristic is higher than or equal to the utility of the outcome for the agent having a UCP-Net, that agent receives one point. Since 10 different opponent agents (*Agent Bs*) negotiate with the same agent (*Agent A*) 10 times, we count the number of times that *Agent A* using a heuristic is at least as successful as *Agent A* having a UCP-net over 100 negotiations. Recall that we have ten different CP-net and UCP-net pairs for *Agent A* in our experiments. When we consider all negotiations, we have 1,000 negotiations between *Agent A* and *Agent B*.
- **Average number of rounds for reaching a consensus:** The duration of the negotiation has been used as a performance metric in a variety of negotiation studies [4, 9, 28]. This metric denotes how fast the agreement is reached. We evaluate each approach with respect

to their speed for a successful negotiation. For this purpose, we report the average number of rounds that is needed to reach a consensus over 100 negotiations. Note that a round is the interaction of an agent with another agent by means of sending an offer—the number of offers that are made by agents. Thus, it is desired to negotiate with fewer rounds.

- **Number of times that Agent A applying a heuristic negotiates at least as fast as Agent A having a UCP-net:** Finally, we observe the number of times that when Agent A applies a particular heuristic and negotiates at least as fast as if it has its user’s real total preference ordering as a UCP-net. If the agent applying a particular heuristic negotiates faster or just as fast as a UCP-net, the agent applying that heuristic receives one point. For each CP-net and UCP-net pair, we evaluate 100 negotiations and compare the scores that the heuristics get at the end.

Our first evaluation criterion is the average utility of negotiation outcomes with respect to Agent A’s preferences over 100 different negotiations with Agent B. Table 2 shows the average utilities of negotiation outcomes for 10 different Agent As. The first column of the table shows which CP-net is used by Agent A, and the remaining five columns indicate the average utility of the negotiation outcomes for Agent A over 100 negotiations with Agent B while this agent is applying a particular heuristic (DL, DLH, PRH, POH, TH, AAH). The seventh column shows the average utility for the case when Agent A uses random utilities, whereas the last column shows it for the case when Agent A has its user’s real total preference ordering as a UCP-net. Note that the performance of DH and DLH is the same for all negotiations because simple concession-based strategy is sensitive to the ordering of the outcomes rather than their extent. Therefore, while we talk about the performance of DH in the remainder of this section, we mean the performance of both heuristics.

As expected Agent A using UCP-net gets the highest score since it negotiates with its user’s real preference orderings. Notice that the performance of all the proposed heuristics is better than the random approach. When we investigate the performance of the proposed heuristics elaborately, it can be seen that Agent A using DH gets the highest average utility in 7 out of 10 CP-nets, whereas Agent A using TH gets the highest average utility in 6 out of 10 CP-nets. Furthermore, Agent A using POH gains the highest average utility in 5 out of 10 CP-nets. Thus, these three heuristics seem to be good candidates in case we only have a user’s CP-net instead of a UCP-net.

Table 2 Average utility of negotiation outcomes over 100 negotiations for Agent A

CP-net	DH–DLH	PRH	POH	TH	AAH	Random	UCP-net
1	0.85	0.88	0.87	0.82	0.85	0.71	0.91
2	0.97	0.95	0.97	0.98	0.97	0.80	0.98
3	0.97	0.94	0.97	0.96	0.96	0.82	0.97
4	0.95	0.95	0.95	0.95	0.95	0.78	0.96
5	0.98	0.98	0.98	0.98	0.98	0.94	0.99
6	0.96	0.95	0.93	0.96	0.95	0.81	0.98
7	0.94	0.92	0.94	0.93	0.94	0.81	0.96
8	0.99	0.98	0.99	0.99	0.99	0.91	0.99
9	0.98	0.96	0.98	0.99	0.98	0.81	0.98
10	0.98	0.96	0.96	0.97	0.97	0.82	0.99
Avg	0.96	0.95	0.95	0.95	0.95	0.82	0.97

Bold represents the highest average utility for a given CP-net

Furthermore, the last row of Table 2 shows the average utility of negotiation outcomes with respect to *Agent A*'s preferences over all negotiations between *Agent As* and *Agent Bs* (10×100 negotiations). Among all heuristics, the performance of DH is slightly higher than the performance of the other heuristics ($0.96 > 0.95$). We have analyzed these 1,000 negotiation results using ANOVA (Analysis of Variance). Note that ANOVA provides a statistical approach to test the differences between more than two related sample means (the performance of the heuristics in our experiments) [19]. Since we have two factors: CP-net and Heuristic, we apply ANOVA two-factor analysis. When we analyze the results for DH with those for POH and TH, we see that the mean of the utilities of negotiation outcomes for *Agent A* using DH is statistically different from the means of the utilities for *Agent A* using POH and TH under 95 percent confidence level where $F > F_{crit}$ and $pvalue < 0.05$ for both factors. That is, it can be said that the average utility of the agent applying DH is higher than that of the agent applying either POH or TH. Notice that the utilities gained with UCP-nets are rather high. From our perspective, the absolute values of utilities are not significant, since our aim is to find heuristics that perform close to UCP-nets' utilities. Hence, our aim is to closely estimate the utilities with respect to UCP-net's utilities.

We also perform the worst-case analysis of the proposed heuristics' performance with respect to *Agent A*'s utility of the negotiation outcomes. To do this, we evaluate the lowest utility that *Agent A* gets over 100 negotiations for each CP-net and UCP-net pair that is shown in Table 3. According to the results in Table 3, it is obvious that the highest utility belongs to *Agent A* negotiating with UCP-nets. The second highest utility belongs to *Agent A* applying DH heuristic in 7 of 10 cases where *Agent A* applying TH gets the second highest utility in 6 of 10 cases. If we consider all of the negotiations (1,000 negotiations), the second highest utility belongs to *Agent A* applying DH. These results also support that the performance of DH is slightly better than other heuristics.

Our second evaluation criterion is the number of times that the agent with a heuristic negotiates at least as well as the agent having a UCP-net. Table 4 involves the evaluation of this criterion for each CP-net over 100 negotiations. The last row of the table shows the number of times that the agent applying a heuristic on 10 CP-nets and negotiates as well as the agent having total preference ordering over 1,000 negotiations. Although the number of times DH performs as well as UCP-net seems to be the highest for only tenth CP-net, it

Table 3 The lowest utility that *Agent A* gets over 100 negotiations

CP-net	DH-DLH	PRH	POH	TH	AAH	Random	UCP-net
1	0.74	0.73	0.73	0.68	0.68	0.62	0.84
2	0.92	0.85	0.86	0.92	0.88	0.77	0.95
3	0.94	0.84	0.94	0.94	0.91	0.77	0.94
4	0.90	0.86	0.88	0.84	0.86	0.65	0.90
5	0.94	0.82	0.94	0.96	0.96	0.73	0.96
6	0.84	0.84	0.84	0.89	0.84	0.79	0.93
7	0.86	0.78	0.86	0.86	0.86	0.71	0.89
8	0.95	0.90	0.96	0.95	0.96	0.72	0.98
9	0.92	0.85	0.92	0.92	0.87	0.69	0.92
10	0.87	0.83	0.86	0.85	0.87	0.74	0.97
Min lowest	0.74	0.73	0.73	0.68	0.68	0.62	0.84

Bold shows the best worst-case performance for the heuristics

Table 4 Number of times heuristics perform as well as UCP-nets over 100 negotiations

CP-net	DH-DLH	PRH	POH	TH	AAH	Random
1	34	44	47	25	36	0
2	44	46	50	68	61	0
3	53	38	59	41	45	0
4	53	50	44	60	58	1
5	64	86	67	53	52	22
6	48	49	32	48	45	0
7	57	37	59	43	45	0
8	89	78	92	93	89	4
9	74	63	75	91	85	2
10	62	48	47	53	49	0
Sum	578	539	572	575	565	29
Percentage	57.8	53.9	57.2	57.5	56.5	2.9

Bold shows the highest scores for the heuristics

is seen that DH has the highest score (578) while considering all negotiation results (1,000 negotiations). This stems from the fact that the variance of DH is less than the other heuristics. In this respect, we can identify DH as the best reliable heuristic.

When *Agent A* uses its user's CP-net and applies DH, it negotiates at least as well as the agent having total preference ordering (UCP-Net) in 57.8 percent of negotiations and *Taxonomic Heuristic* (TH) is successful at least as UCP-Net in 57.5 percent of negotiations. Also, the performance of *Preferred Outcomes Heuristic* (POH) is almost near to DH and TH (57.2). As far as the performance of the heuristics with respect to the first two evaluation criteria is concerned, DH, TH and POH might be considered as alternative heuristics that the user may use in negotiation when it has a CP-net. The performance of DH is slightly better than others so it may be preferred over TH and POH.

Next, we investigate how fast *Agent A* negotiates. Table 5 shows the average number of rounds to reach a consensus over 100 negotiations. Note that we do not investigate the average number of rounds for Random since its performance for our first two criteria is quite insufficient. On average the agent using DH negotiates at 24 rounds while the agent using POH negotiates at 33 rounds. When we apply ANOVA two-factor analysis, it can be said that the mean of DH is statistically different from the mean of POH under 95 percent confidence level where $F > F_{crit}$ and $p_{value} < 0.05$ for both factors. Under the same confidence level, the mean of DH is also statistically different from the mean of UCP-net. As a result, it can be said that the agent having a CP-net and applying DH negotiates faster than the agent having a UCP-net and negotiating with the user's real total preference ordering. Further, observe that the agent that uses DH heuristic reaches agreements faster than any other approach. In this respect, DH has a clear advantage compared to all other approaches, and thus it can be concluded that it is best to use a UCP-net but when this is not available, a CP-net with DH heuristic is the best trade-off which negotiates faster.

We also investigate the worst-case analysis of *Agent A*'s performance with respect to the number of rounds that are required to complete a negotiation session successfully. Table 6 shows the highest number of rounds to reach a consensus over 100 negotiations for each CP-net and UCP-net pair. According to the results, *Agent A* applying DH negotiates faster than others in 7 out of 10 cases where others negotiate faster individually in only 1 out of 10 cases. At overall, it can be said that DH's worst-case performance is better than others.

Table 5 Average number of rounds to reach a consensus

CP-net	DH–DLH	PRH	POH	TH	AAH	UCP-net
1	30.07	65.02	39.78	55.62	49.99	68.96
2	33.23	65.75	44.43	78.68	73.91	56.63
3	24.18	38.90	37.28	39.81	50.06	52.06
4	27.97	32.12	29.87	37.99	27.27	68.36
5	8.48	21.97	10.10	17.25	20.13	21.30
6	22.43	24.82	27.28	25.63	21.13	35.59
7	22.82	31.28	31.58	36.34	37.79	29.33
8	11.4	14.94	14.77	16.47	15.66	14.47
9	39.88	43.96	43.74	48.14	37.31	47.21
10	21.88	53.68	49.29	45.25	44.22	38.09
AVG	24.23	39.24	32.81	40.12	37.75	43.20

Bold shows the lowest average number of rounds for a given CP-net

Table 6 The highest number of rounds to reach a consensus over 100 negotiations

CP-net	DH–DLH	PRH	POH	TH	AAH	UCP-net
1	131	500	215	500	500	500
2	141	500	500	500	500	419
3	255	431	363	500	485	500
4	235	202	373	500	285	500
5	129	277	129	277	339	339
6	419	197	419	225	195	500
7	253	387	387	413	465	253
8	37	113	139	87	81	97
9	493	313	493	500	236	485
10	107	500	487	500	500	500

Bold shows the best worst-case performance for a given CP-net

We also evaluate the number of times that the agent having a CP-net and applying a particular heuristic negotiates at least as fast as the agent having a total ordering of outcomes as a UCP-net. Table 7 shows the results for this criterion. According to this result, it is seen that DH and POH outperforms other heuristics in terms of this criterion. In 68.1 percent of negotiations, the agents applying DH negotiates at least as fast as the agents using UCP-nets while the agents applying POH negotiates at least as fast as the agents using UCP-nets in 67.5 percent of negotiations. On the other hand, agents using other heuristics negotiates at least as fast as the agent using UCP-nets in about 63 percent of negotiations. This result is a confirmation of our previous result shown in Tables 5 and 6.

As a result, the agent may prefer to apply *Depth Heuristic* or *Preferred Outcome Heuristic* in terms of “success” and “duration” when it has a CP-net. Since the performance of DH is slightly better than POH, DH may be preferred over POH. Moreover, when we consider the implementation details and computational complexity of the proposed heuristics, DH can be identified as the leading heuristic. In *Depth Heuristic*, we consider only the depth of an outcome and the depth of preference graph while in *Preferred Outcome Heuristic* we take all

Table 7 Number of times heuristics negotiate at least as fast as UCP-net

Cp-net	DH-DLH	PRH	POH	TH	AAH
1	67	57	61	66	60
2	48	44	51	40	45
3	64	64	62	70	63
4	72	73	78	76	75
5	89	51	88	79	82
6	69	80	76	76	77
7	59	69	55	59	55
8	70	74	71	45	52
9	70	47	68	57	60
10	73	68	65	61	57
Sum	681	627	675	629	626
Percentage	68.1	62.7	67.5	62.9	62.6

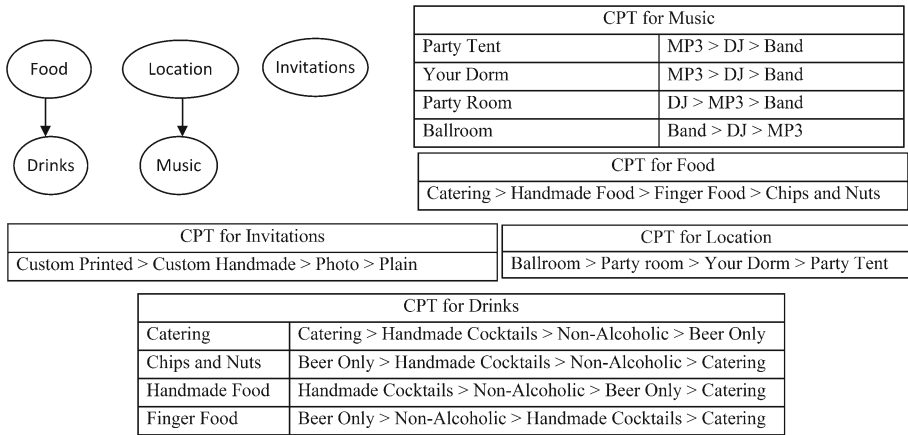


Fig. 5 The CP-net for *Agent A* in our party domain

of the outcomes which is preferred over the current outcome (requiring search through the graph) and depth of those outcomes into consideration. The former heuristic seems to be more simple. Concluding we recommend to apply DH when an agent has its user’s preferences as a CP-net for the strategy considered here.

4.5 Additional experiments

Apart from the user experiments, we have run additional experiments in another domain to validate the results. In this setup, we have chosen a party domain including five issues: *Location, Food, Drink, Music, and Invitation*. This domain has 768 alternative outcomes where the holiday domain has 216 outcomes. For *Agent A*, we have created the CP-net preference profile shown in Fig. 5. To simulate three different participants for *Agent A* having the same partial ordering (CP-net) but different total orderings (UCP-nets), we have constructed three con-

sistent UCP-net profiles with the given CP-net. For *Agent B*, we have generated five different preference profiles. In our experiments, each *Agent A* negotiates with each *Agent B* 20 times.

In this experiment setup, we take into account the tradeoff between the computational cost of the heuristics and their performance in negotiation. In the user experiments, we have observed that the computational cost of POH and AAH is significantly higher than others; however, their overall performance is not much better than others. Therefore, we do not use those heuristics in this experiment. First of all, we investigate how well *Agent A* negotiates with its opponents. Table 8 shows the average utility of negotiation outcomes for *Agent A* over 100 negotiations. Recall that the utility of the negotiation outcomes is estimated according to the true total ordering represented by means of UCP-nets. As expected, *Agent A* using UCP-net gains higher utility on average since it has the agent's real total ordering. For the first case where *Agent A*'s true preferences are represented by the first UCP-net, TH performs slightly better than DH while DH performs slightly better than TH for the third case. On average, their performance is better than PRH.

Table 9 shows how many times the given heuristic performs at least as well as UCP-net. In 56 and 57 % of negotiations, DH and TH perform at least as well as UCP-net, respectively, while PRH performs at least as well as UCP-net in only 39 % of negotiations. Those results also support that the performance of DH and TH is almost the same and better than PRH.

Finally, we study how fast the agents reach an agreement. Table 10 shows how many times an agent with a given heuristic negotiates at least as fast as an agent with a UCP-net. According to those results, DH outperforms other heuristics. If we consider both how well the agent with the given heuristic negotiates and how fast that the agreement is reached, DH heuristic is to be the best trade-off. As a result, those results support the results of the user experiments.

Without doubt, we cannot guarantee that DH heuristic will always outperform the others. The negotiation outcome depends on a variety of factors such as domain size, the number and complexity of the dependencies in the given CP-net, the complexity of the preference graph induced from the CP-net and the degree of the conflict between agents. Therefore, the given results only reflects an approximation of the relative performance of the heuristics in negotiation.

Table 8 Average utility of negotiation outcomes over 100 negotiations for *Agent A* in party domain

Case	DH	PRH	TH	Random	UCP-net
UCP-net 1	0.864	0.855	0.876	0.504	0.877
UCP-net 2	0.881	0.852	0.885	0.585	0.902
UCP-net 3	0.942	0.931	0.939	0.756	0.940

Bold represents the highest average utility for a given CP-net

Table 9 Number of times heuristics perform as well as UCP-nets over 100 negotiations in party domain

Case	DH	PRH	TH	Random
UCP-net 1	56	40	66	0
UCP-net 2	45	26	46	0
UCP-net 3	66	52	59	0
Sum	167	118	171	0
Percentage	55.67	39.33	57.00	0.00

Bold shows the highest scores for the heuristics

Table 10 Number of times heuristics negotiate at least as fast as UCP-net in party domain

CASE	DH	PRH	TH
UCP-net 1	72	59	69
UCP-net 2	56	43	49
UCP-net 3	66	45	51
Sum	194	147	169
Percentage	64.67	49.00	56.33

5 Related work and discussion

In this section, we review related work from the literature. We first overview a variety of negotiation approaches in computer science and then review a variety of ranking approaches for alternatives when we have partial preference ordering of alternatives as CP-nets or CP-net-like models.

5.1 Some negotiation approaches

Luo et al. [31] propose a prioritized fuzzy constraint-based negotiation model in which the buyer agent specifies its offers using constraints. According to the proposed model, the agent reveals its partial preference information (some parts of its constraints) in a minimal fashion. That is, the buyer agent sends its constraints to the seller. The seller agent checks the constraints and if it cannot satisfy these constraints, it requests the buyer to relax them. In our case, instead of revealing the agent's partial preferences to the opponent, the agent tries to negotiate on its user's behalf while it has only its user's partial preference information represented as a CP-net.

Aydođan and Yolum also use the constraints to model the users' preferences in an automated negotiation setting [4]. In their setting, the producer agent tries to understand and predict the consumer agent's preferences represented as conjunctive and disjunctive constraints, and accordingly generate well-targeted offers. Similarly, Jonker, Robu and Treur propose a guessing heuristic for bilateral multi-issue negotiation in which the negotiating agent has a partial knowledge about its opponent preferences represented as linear additive utility functions [27]. Thus, the agent tries to predict the unknown weights using the history of its opponent's bids. The guessing heuristic is based on attribute value distance between the opponent's bids. Furthermore, Coehoorn and Jennings extend Faratin's trade-off strategy [15] by incorporating the prediction of the opponent's weights [11]. To predict the opponent's weights, they apply a nonparametric statistical method, namely kernel density estimation. Their intuition is that the difference between the opponent's last offers may have a relation with the opponent's weights. Instead of focusing on computer agents' belief about others in negotiation, Ficici and Pfeffer analyze the human-subject's behavior in a negotiation game and present a model of human reasoner's belief about others [17]. Contrary to these studies, in our case, the agent has partial preference information about its own users. Instead of predicting opponent's unknown preference parameters, we focus on developing heuristics to estimate a total ordering of outcomes from a partial ordering represented via a CP-net to negotiate effectively.

Chalamish and Kraus present an automated mediator for bilateral negotiations in which agents share their qualitative preferences only with the mediator [9]. The aim of the mediator is to suggest acceptable agreements for both participants when it is necessary to speed

up negotiations. In the proposed system, each agent represents its user's preferences as a weighted CP-net, an extension of a CP-net with weighted importance table indicating the relative importance of attributes. After both agents send their WCP-nets to the mediator, the mediator sorts all possible outcomes with respect to agents' preferences separately—resulting in two sorted list of outcomes. While sorting outcomes, the mediator uses an enhanced version of majority lexicographical (ML) ordering [33] with weights. The weighted version of ML presented by Chalamish and Kraus is based on the comparison of weighted sum that the outcome gets for its assignment of each attribute. For instance, assume that for X attribute it is specified that $x_1 > x_0$, which means x_1 is preferred over x_0 and the relative importance of X is expressed with $riw(X)$. If the outcome involves x_1 , it gets $riw(X)$ while it gets zero if it includes x_0 . When we compare two outcomes, α and β , if the summation for α is higher than that for β , it is said that α is preferred over β . To sum up, the mediator sorts outcomes with respect to this metric and recommends Pareto-optimal outcomes.

While the mediator uses other agents' WCP-Nets to suggest Pareto-optimal outcomes to the negotiating agents, in our study the agent tries to derive estimated utility values from its own CP-net by applying one of the proposed heuristics. These estimated utility values will be employed by the agent's negotiation strategy while negotiating with the opponent agent. Since we do not have relative importance weights, it is hard to compare our approaches with theirs directly. However, it can be said that it is straightforward to apply their approach in binary WCP-nets but it is not clear how the rank of each outcome will be estimated when the attributes have more than two values.

5.2 Some ranking approaches

Rossi et al. [34] extend CP-nets to capture multiple agents' qualitative preferences and present *mCP-nets*. In *mCP-nets*, there exists a partial CP-net for each individual agent. In contrast to CP-nets, partial CP-nets allow users not to specify the ordering of the values of some particular attributes. This is necessary to be able to represent preferences in *mCP-nets* such as "My preference on x depends on other agent's preference on x ". Thus, we have additional flips such as indifferent and incomparable flips in addition to improving/worsening flips. When there is a chain of worsening and indifferent flips and at least one worsening flip from α and β , it is said that α is preferred over β . Unlike CP-nets, there may be more than one optimal outcome in an acyclic partial CP-net because of the unranked attributes.

In that study, several voting methods such as Pareto, Majority, Max, Lex and Rank [34] are proposed to aggregate a group of agents' preferences and to determine whether an outcome is preferred over another for that group of agents. Among these semantics, only *Rank* removes incomparability. There may be some incomparable outcomes with respect to other semantics except *Rank*. According to *Rank* method, each agent ranks each outcome by estimating the length of the shortest sequence of worsening flips between that outcome and one of the optimal outcomes. While comparing two outcomes, each agent's rank is summed up for these outcomes. To be able to say that an outcome α is preferred over β , the sum of ranks assigned to α should be smaller than the sum of ranks for β .

Unlike that study, we do not address the problem of capturing and handling the qualitative preferences of a group of agents. Instead, we focus on how one agent reasons on its user's qualitative preferences represented as an acyclic CP-net and negotiates accordingly with an opponent agent. In our depth-based heuristic, we rank an outcome by taking into account the longest sequence of improving flips between the worst outcome and that outcome. Rossi et al. on the other hand, consider the shortest sequence of worsening flips to find the optimal outcome.

We briefly examine the differences in our and their rankings by considering the CP-net explained in Example 2 and the preference graph induced from this CP-net depicted in Fig. 3. According to their *Rank* method, $\langle \text{Sea, Winter, Two weeks} \rangle$ is preferred over $\langle \text{Historical Places, Winter, One week} \rangle$ since the length of the shortest sequence of worsening flips between the optimal outcome, $\langle \text{Sea, Summer, Two weeks} \rangle$, and $\langle \text{Sea, Winter, Two weeks} \rangle$ is equal to one while that between the optimal outcome and $\langle \text{Historical Places, Winter, One week} \rangle$ is equal to three. Recall that the smaller rank is accepted as more preferable with respect to *Rank* method. However, the length of the longest sequence of improving flips from the worst outcome to the current outcome is the same for both outcomes. In other words, they are at the same depth (five). Therefore, these two outcomes are equally preferred with respect to our depth heuristic.

Li et al. [29] also study the problem of collective decision making with CP-nets. The aim of that study is to find a Pareto-optimal outcome when agents' preferences represented by CP-nets. They firstly generate candidate outcomes to increase the computational efficiency instead of using the entire outcome space. Then, each agent in the system ranks these candidate outcomes according to their own CP-nets. For ranking an outcome, they use the longest path between the optimal outcome and that outcome in the induced preference graph. Thus, the minimum rank is desired for the agents. They choose the final outcome for the agents by minimizing the maximum rank of the agents.

While they propose a procedure for collective decision making with the aim of choosing one outcome for multiple agents, we focus on estimating utility values of each outcome that will be used during the negotiation for an individual agent. For this purpose, we use the longest path between the worst outcome and that outcome, which corresponds to the depth of an outcome in the preference graph, to estimate the utilities with our depth heuristic.

Rossi and Venable propose two new ways of reasoning with CP-nets: applying new semantics for CP-nets and approximating CP-nets with soft constraint satisfaction problem [33]. Similar to our approach, the new semantics provide a total order while the original CP-net semantics provides a partial one. The proposed approaches in their study takes the hierarchy level of each attribute in CP-nets into account while comparing outcomes. The attributes are ordered with respect to their dependencies where the independent attributes take the first places. According to the first semantics, α is preferred over β if α dominates β first lexicographically. To illustrate this, consider we have two attributes R and S and our CP-net says $r_1 \succ r_0$, $r_1 : s_1 \succ s_0$ and $r_0 : s_0 \succ s_1$. Note that R is in the first level where S is the second level. When we compare two outcomes: $\langle r_1, s_0 \rangle$ and $\langle r_0, s_0 \rangle$, we say that the former outcome is preferred over the latter outcome because it has a better value (r_1) for the first attribute (implicitly more important than the second attribute).

In the second semantics, a so-called *majority lexicographical (ml) ordering*, the outcome dominating on the majority of the important attributes (that are in the higher hierarchical level) is more preferred. To illustrate this, assume that we have five attributes R, S, X, Y and Z and a CP-Net saying that $r_1 \succ r_0$, $s_1 \succ s_0$, $s_1 : x_1 \succ x_0$, $s_0 : x_0 \succ x_1$, $s_1 : y_1 \succ y_0$, $s_0 : y_0 \succ y_1$, $x_1 : z_1 \succ z_0$ and $x_0 : z_0 \succ z_1$. In this CP-net, R and S are in the first level, X and Y are in the second level and Z is the third level. According to ML ordering, the outcome $\langle r_1, s_0, x_0, y_0, z_1 \rangle$ dominates $\langle r_0, s_1, x_1, y_0, z_1 \rangle$. Since the former outcome wins on R and the latter outcome wins on S , they tie on the first level. However, in the second level, the former outcome wins on both X and Y while the latter outcome only wins on X . Since $\langle r_1, s_0, x_0, y_0, z_1 \rangle$ wins on the majority in the second level, it dominates $\langle r_0, s_1, x_1, y_0, z_1 \rangle$. Note that it is not needed to check for the attribute in the third level. In our study, we use the induced preference graph to find the estimated utilities of the outcomes

and compare outcomes with respect to these estimated utilities while they use the hierarchical level of CP-net directly to compare two outcomes.

Given that the results presented here prove the potential of using CP-nets or other forms of partial orderings of preference, it now becomes a point of future research to determine what is the best ranking approach to use.

6 Conclusion

In this paper, we have investigated the ways of using acyclic CP-nets with the negotiation strategies employing quantitative information such as utilities. Thus, we have presented several heuristics to produce estimated utilities from acyclic CP-nets. These estimated utilities are used by the agent's negotiation strategy in order to generate offers or to decide whether the opponent's counter offer is acceptable during the negotiation. In this way, the agent can still apply a negotiation strategy using quantitative information such as utilities while it has its user's CP-net. Here, our aim is to map qualitative preferences to quantitative preferences.

To compare the performance of our heuristics in a realistic setting, we have extended the GENIUS negotiation framework. The framework allows the user to express her preferences both quantitatively (UCP-nets) and qualitatively (CP-nets). The quantitative model namely UCP-net serves as a ground truth. All negotiation outcomes are evaluated with respect to the given UCP-net. Thus, we have compared the performance of agents when they apply heuristics on their CP-nets and negotiate with the estimated utilities versus when they have their user's real total preference ordering as UCP-nets and negotiates with the real utilities. It is worth noticing that CP-nets can tolerate partial ordering, which means there are some incomparable outcomes with respect to the given preferences. By applying a heuristic, we induce an estimated total ordering from a partial ordering.

We have evaluated the performance of the heuristics by taking into account the average utility of the negotiation outcomes and the number of rounds that are required to reach a consensus. As expected, the agent using user's real total ordering (UCP-nets) performs best with respect to overall utility gained. However, our results show that it is admissible to elicit the user's preferences as a CP-net and to negotiate with these preferences by applying a heuristic instead of eliciting user's preferences quantitatively. Among the proposed heuristics, Depth Heuristic can be identified as the leading heuristic. That is, the agent may prefer to apply Depth Heuristic in terms of both success and speed when it elicits its user's preferences as an acyclic CP-net.

Our results indicate that using heuristics with CP-nets reveal a successful mechanism for negotiation. However, as expected, we have also realized that in a few worst-case situations (see Table 3, CP-net 1) all of the heuristics perform significantly worse than the UCP-net. It would be interesting to investigate reasons behind this. Another direction for future research is studying the effect of negotiation strategies on the performance of the heuristics. In this study, the agent applies a simple concession-based strategy that is explained in Sect. 4.3. It would be interesting to apply other negotiation strategies such as Boulware and compare their effects. It would be ideal to identify pairings of heuristics and strategies that perform well together.

Acknowledgments This research has been supported by Boğaziçi University Research Fund under grant BAP5694 and the Scientific and Technological Research Council of Turkey. Most of this work has been done when Reyhan Aydoğan was at Bogazici University. Some of the ideas presented in this paper initially appeared in [1–3].

Table 11 Information about CP-nets and their induced preference graphs

CP-net	Dependency	Hierarchy	Independent nodes	Orderings	Graph depth
1	1	2	4	7	10
2	5	3	1	15	16
3	4	3	2	16	16
4	4	2	1	15	18
5	4	3	1	12	18
6	3	3	2	10	13
7	3	3	1	11	14
8	5	3	1	17	22
9	4	3	1	13	21
10	5	3	1	16	19

Appendix: CP-nets used in our experiments

Table 11 gives information about 10 users' CP-nets and induced preference graphs from these CP-nets. The second column shows how many dependencies exist in the CP-net, which is equal to the number of edges in the CP-net. Note that more dependency may express more information about user's preferences. The third column indicates the level of hierarchy—the length of the longest path between ancestor and descendant nodes. This table also shows the number of independent nodes (not having any connections with other nodes) in the CP-net and total number of orderings expressed in CPTs. The last column indicates the depth of induced preference graph from this CP-net.

References

1. Aydoğan R, Baarslag T, Hindriks K, Jonker CM, Yolum P (2013) Heuristic-based approaches for CP-nets in negotiation. In: Ito T, Zhang M, Robu V, Matsuo T (eds) *Complex automated negotiations: theories, models, and software competitions*, vol 435 of *studies in computational intelligence*, pp 113–123
2. Aydoğan R, Tasdemir N, Yolum P (2010) Reasoning and negotiating with complex preferences using CP-nets. In: Aalst W, Mylopoulos J, Sadeh NM, Shaw MJ, Szyperski C, Ketter W, Poutr H, Sadeh N, Shehory O, Walsh W (eds) *Agent-mediated electronic commerce and trading agent design and analysis*, vol 44 of *lecture notes in business information processing*, pp 15–28
3. Aydoğan R, Yolum P (2010) Effective negotiation with partial preference information. In: *Proceedings of the ninth international joint conference on autonomous agents and multiagent systems (AAMAS)*. Toronto, Canada, pp 1605–1606
4. Aydoğan R, Yolum P (2012) Learning opponents preferences for effective negotiation: an approach based on concept learning. *J Auton Agents Multi-Agent Syst* 24(1):104–140
5. Baarslag T, Hindriks K, Jonker CM, Kraus S, Lin R (2010) The first automated negotiating agents competition (ANAC 2010). In: Ito T, Zhang M, Robu V, Fatima S, Matsuo T (eds) *New trends in agent-based complex automated negotiations*, series of *studies in computational intelligence* (to appear). Springer, Berlin
6. Boutilier C, Bacchus F, Brafman RI (2001) UCP-networks: a directed graphical representation of conditional utilities. In: *Proceedings of the seventeenth annual conference on uncertainty in artificial intelligence*. Seattle, USA, pp 56–64
7. Boutilier C, Brafman RI, Domshlak C, Hoos HH, Poole D (2004) CP-nets: a tool for representing and reasoning with conditional ceteris paribus preference statements. *J Artif Intell Res* 21:135–191
8. Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. *Comput Netw ISDN Syst* 30(1–7):107–117

9. Chalamish M, Kraus S (2007) Automated: an automated mediator for bilateral negotiations under time constraints. In: Proceedings of the sixth international joint conference on autonomous agents and multiagent systems, pp 248:1–3
10. Chen S, Weiss G (2012) A novel strategy for efficient negotiation in complex environments. In: Timm JJ, Guttman C (eds) 'MATES', vol 7598 of lecture notes in computer science. Springer, Berlin, pp 68–82
11. Coehoorn RM, Jennings NR (2004) Learning on opponent's preferences to make effective multi-issue negotiation trade-offs. In: Proceedings of the sixth international conference on electronic commerce, pp 59–68
12. Dan Pelleg AM (2000) X-means: extending K-means with efficient estimation of the number of clusters. In: Proceedings of the seventeenth international conference on machine learning. Morgan Kaufmann, San Francisco, pp 727–734
13. Domshlak C, Prestwich SD, Rossi F, Venable KB, Walsh T (2006) Hard and soft constraints for reasoning about qualitative conditional preferences. *J Heuristics* 12(4–5):263–285
14. Faratin P, Sierra C, Jennings NR (1998) Negotiation decision functions for autonomous agents. *Int J Robot Auton Syst* 24(3–4):159–182
15. Faratin P, Sierra C, Jennings NR (2002) Using similarity criteria to make issue trade-offs in automated negotiations. *Artif Intell* 142:205–237
16. Fatima S, Wooldridge M, Jennings N (2007) On efficient procedures for multi-issue negotiation. In: Fasli M, Shehory O (eds) Automated negotiation and strategy design for electronic markets, vol 4452 of lecture notes in computer science. Springer, Berlin, pp 31–45
17. Ficici SG, Pfeffer A (2008) Simultaneously modeling human's preferences and their beliefs about other's preferences. In: Proceedings of the 7th international joint conference on autonomous agents and multiagent systems-vol 1, AAMAS '08, pp 323–330
18. Fishburn PC (1970) Utility theory for decision making. Wiley, New York
19. Fisher RA (1959) Statistical methods and scientific inference, 2nd edn. Hafner Publishing Co., New York
20. Gode DK, Sunder S (1993) Allocative efficiency of markets with zero-intelligence traders: market as a partial substitute for individual rationality. *J Polit Econ* 101(1):119–137
21. Hao J, Leung H-F (2012) Abines: an adaptive bilateral negotiating strategy over multiple items. In: Proceedings of the the 2012 IEEE/WIC/ACM international joint conferences on web intelligence and intelligent agent technology, vol 02', WI-IAT '12. IEEE Computer Society, Washington, DC, USA, pp 95–102
22. Hindriks K, Jonker CM, Kraus S, Lin R, Tykhonov D (2009) Genius: negotiation environment for heterogeneous agents. In: Proceedings of the eighth international joint conference on autonomous agents and multiagent systems (AAMAS), vol 2, pp 1397–1398
23. Hindriks K, Tykhonov D (2008) Opponent modelling in automated multi-issue negotiation using bayesian learning. In: Proceedings of the seventh international joint conference on autonomous agents and multiagent systems (AAMAS). Estoril, Portugal, pp 331–338
24. Hindriks KV, Tykhonov D (2010) Towards a quality assessment method for learning preference profiles in negotiation. In: Aalst W, Mylopoulos J, Sadeh NM, Shaw MJ, Szyperki C, Ketter W, Poutre H, Sadeh N, Shehory O, Walsh W (eds) Agent-mediated electronic commerce and trading agent design and analysis, vol 44 of lecture notes in business information processing, pp 46–59
25. Jain R, Purohit DGN (2011) Page ranking algorithms for web mining. *Int J Comput Appl* 13(5):22–25
26. Jennings NR, Faratin P, Lomuscio AR, Parsons S, Sierra C, Wooldridge M (2001) Automated negotiation: prospects, methods and challenges. *Int J Group Decis Negot* 10(2):199–215
27. Jonker CM, Robu V, Treur J (2007) An agent architecture for multi-attribute negotiation using incomplete preference information. *JAAMAS* 15(2):221–252
28. Klos T, Somefun K, Poutre HL (2011) Automated interactive sales processes. *IEEE Intell Syst* 26(4):54–61
29. Li M, Vo QB, Kowalczyk R (2010) An efficient procedure for collective decision-making with CP-nets. In: Proceeding of the 19th European conference on artificial intelligence, Lisbon, Portugal, pp 375–380
30. Lin R, Kraus S, Baarslag T, Tykhonov D, Hindriks K, Jonker CM (2014) Genius: an integrated environment for supporting the design of generic automated negotiators. *Comput Intell* 30(1):48–70
31. Luo X, Jennings NR, Shadbolt N, Leung H, Lee JH (2003) A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. *Artif Intell* 148(1–2):53–102
32. Raiffa H (1982) The art and science of negotiation. Harvard University Press, Cambridge
33. Rossi F, Venable KB, Walsh T (2002) CP-networks: semantics, complexity, approximations and extensions. In: Proceedings of the fourth international workshop on soft constraints (Soft-02) in CP-2002, USA
34. Rossi F, Venable KB, Walsh T (2004) mCP nets: representing and reasoning with preferences of multiple agents. In: Proceeding of the nineteenth national conference on artificial intelligence, pp 729–734
35. Russell SJ, Norvig P (1995) Artificial intelligence: a modern approach. Prentice Hall, Englewood Cliffs

36. Sierra C, Faratin P, Jennings N (1997) A service-oriented negotiation model between autonomous agents. In: Boman M, van de Velde W (eds) Proceedings of the eighth European workshop on modelling autonomous agents in multi-agent world, MAAMAW'97', vol 1237 of lecture notes in artificial intelligence. Springer, Berlin, pp 17–35



Reyhan Aydoğan is a postdoctoral researcher in the Interactive Intelligence group at Delft University of Technology, the Netherlands. She received her M.Sc. and Ph.D. degrees from the Department of Computer Engineering at Boğaziçi University and her B.S. degree from the Department of Computer Engineering at Dokuz Eylül University. During her M.Sc. and Ph.D. studies, Dr. Aydoğan was funded by the Scientific and Technological Research Council of Turkey (TUBITAK) under the national M.Sc. and Ph.D. student scholarship program. Her research interests are agreement technologies, particularly negotiation, decision support systems, multiagent systems, machine learning, service oriented architectures and semantic Web. She serves as a program committee member in reputable conferences such as AAMAS, ECAI and IJCAI, and a reviewer in highly referred journals such as JAAMAS and JAIR.



Tim Baarslag is a research fellow in the Agents, Interaction and Complexity group at the University of Southampton, where he works on devising techniques to obtain meaningful consent from users in the digital domain. He obtained his PhD at Delft University of Technology on the topic of intelligent decision support systems for automated negotiation. His present research interests include coordination and cooperation in artificial intelligence, privacy and consent, and decision making under uncertainty.



Koen V. Hindriks (Ph.D., Computing Science, Utrecht University) is assistant professor in the Interactive Intelligence section at Delft University of Technology. His main research interests are autonomous decision making using cognitive agents, teamwork and coordination models for multi-agent systems, and human-robot interaction. His current work involves the design and development of the language GOAL that facilitates the engineering of high-level decision making agents. He is a member of the board of the Benelux Association for Artificial Intelligence.



Catholijn M. Jonker (1967) is head of the Interactive Intelligence section at Delft University of Technology. She studied computer science, and did her PhD studies at Utrecht University. She is a member of the Koninklijke Hollandse Maarschappij der Wetenschappen and of the Academia Europaea. She is the president of the National Network Female Professors (LNVH) in The Netherlands, since September 2013. Her research is on cognitive processes and concepts such as trust, negotiation, teamwork and the dynamics of individual agents and organizations. Her mission is to create synergy between humans and technology by understanding, shaping and using fundamentals of intelligence and interaction.



Pinar Yolum holds a Ph.D. in computer science from North Carolina State University, Raleigh, NC, USA. She is currently an Associate Professor of computer engineering in Bogazici University, Istanbul, Turkey, where she leads a research group in multiagent systems that works on trust, agent communication, and social networks. She has (co)authored more than 50 papers in selected journals and conferences. She serves on the Editorial Board of Journal of Autonomous Agents and Multiagent Systems and Journal of AI Research. She is a Board Member of the International Foundation for Autonomous Agents and Multiagent Systems. She serves as the General Co-chair of International Conference on Autonomous Agents and Multiagent Systems for 2015.