Heuristic-Based Approaches for CP-Nets in Negotiation

Reyhan Aydoğan, Tim Baarslag, Koen V. Hindriks,
Catholijn M. Jonker, and Pınar Yolum

Abstract. CP-Nets have proven to be an effective representation for capturing preferences. However, their use in multiagent negotiation is not straightforward. The main reason for this is that CP-Nets capture partial ordering of preferences, whereas negotiating agents are required to compare any two outcomes based on the request and offers. This makes it necessary for agents to generate total orders from their CP-Nets. We have previously proposed a heuristic to generate total orders from a given CP-Net. This paper proposes another heuristic based on Borda count, applies it in negotiation, and compares its performance with the previous heuristic.

1 Introduction

Modeling users' preferences is an inevitable part of automated negotiation tools. While representing the user's preferences, there are several issues to be taken into account. One, outcome space grows exponentially with the number of attributes and their possible values. It may be infeasible to ask a user to rank all outcomes when the outcome space is very large. Two, the user may have difficulty in assessing her preferences in a quantitative way [5]. Representing someone's preferences with numerical values is an arduous task for a human. Three, it is difficult to find a mathematical model for representing preferences in which there are preferential dependencies between attributes. Therefore, it is more effective and intuitive to use a qualitative preference model.

Reyhan Aydoğan · Tim Baarslag · Koen V. Hindriks · Catholijn M. Jonker
Interactive Intelligence Group, Delft University of Technology, Delft, The Netherlands
e-mail: {R.Aydogan,T.Baarslag,K.V.Hindriks}@tudelft.nl
C.M.Jonker@tudelft.nl

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

T. Ito et al. (Eds.): Complex Automated Negotiations, SCI 435, pp. 113–123.
springerlink.com

Although it is desired for users to express their preferences qualitatively, most of the current negotiation strategies [2, 6, 7, 8, 9] work with quantitative preferences. Hence, to use qualitative preferences in negotiation, it is necessary to estimate quantitative preferences from qualitative preferences. Accordingly, this paper is about estimation of quantitative preferences from qualitative preferences. That is, we propose heuristics to allow agents to have a qualitative preference model, while their negotiation strategy employs quantitative information. In order to do so, we start from a qualitative preference representation, namely CP-Nets. CP-Nets allow representation of conditional preferences and tolerate partial ordering. We extend the GENIUS negotiation framework [11] to allow elicitation of acyclic CP-Net preferences. Then, we apply our heuristics to generate utility-based information from the given CP-Net.

We compare the performance of agents when they apply heuristics on their users' qualitative preferences and negotiate with estimated utilities versus when they have their users' real total preference orderings and negotiate with real utilities. To accomplish this, users were asked to create their preference profiles both quantitatively (UCP-Nets) and qualitatively (CP-Nets), using the GENIUS interface for an apartment renting domain. The given UCP-Nets serve as ground truth. The agents apply heuristics on the given CP-Net and then negotiate with the resulting estimated utilities. Each negotiation outcome is evaluated based on the given UCP-Net, which is not only consistent with the CP-Net but also provides a total ordering of outcomes.

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

2 Background: CP-Nets and UCP-Nets

Conditional preference networks (CP-nets) is a graphical model for representing qualitative preferences in a compact way [5]. In CP-nets, each node represents an attribute and each edge denotes preferential dependency between nodes. If there is an edge from X to Y, X is called "parent node" and Y is called "child node". The preference on 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 apartment renting domain in Example 1 and a CP-NET expressing that its user's preference on parking area depends on neighborhood. CPT for *Parking Area* shows that the user prefers an apartment having a parking area when the neighborhood is either *Kadikoy* or *Kartal*. However, she prefers an apartment not having a parking area when it is at *Etiler*. In CP-nets, each preference statement is interpreted under "everything else being equal" interpretation. The statement, "*Etiler* is preferred over *Kartal*", means that if all other attributes such as price and parking area are the same, an apartment at *Etiler* is preferred over an apartment at *Kartal*.

Example 1. For simplicity, we have only three attributes in our apartment renting domain: *Price*, *Neighborhood* and *Parking Area*. There are three neighborhoods: *Etiler*, *Kadikoy* and *Kartal* whereas the valid values for the price are categorized as *High*, *Medium* and *Low*. A parking area may exist or not. Thus, the domain for parking area has two values: *Yes* and *No*.

We need to check whether there exists an *improving flip* sequence from one outcome to another (and vice versa) to answer whether an outcome would be preferred over another. An improving flip is changing the value of a single attribute with a more desired value by using CPT for the attribute. If there are not any improving flip sequences between two outcomes, we cannot compare these two outcomes. Thus, the inability of comparing some outcomes is the challenge of using CP-Nets in negotiation.

Boutilier *et al.* propose UCP-nets [4] by CP-Nets with generalized additive models. UCP-nets are able to represent preferences quantitatively rather than representing simply preference ordering.

Similar to CP-nets, we firstly specify preferential dependency among attributes. Instead of specifying a total preference ordering over the values of each attribute according to their parents' values (conditions), we assign a real value (utility) for all values of each attribute. Utility function $u(X_1, X_2,...X_n)$ is represented in Equation 1 where X_i is the i^{th} attribute of outcome, U_i denotes parents of X_i and $f_i(X_i, U_i)$ represents a factor. Assume that our UCP-Net involves three factors f_1(Neighborhood), f_2(Price) and f_3(Parking Area, Neighborhood). The utility of an outcome is estimated as the sum of these factors.

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

3 Proposed Heuristics

Most of the negotiation strategies [6, 9] work with quantitative preferences such as *utility functions*. However, it is desired for users to express their preferences qualitatively. Thus, we propose heuristics to use acyclic CP-Nets 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 predicted utilities from a given CP-Net by applying our heuristics.

In our framework, a preference graph is induced from a given CP-Net while eliciting a user's preferences as a CP-Net. In this preference graph, each node denotes a possible outcome and each edge represents an improving flip. The direction of edges are ordered from less desired to more desired services. Therefore, the worst outcome will be placed at the top of preference graph (root node) whereas the leaf node holds the best outcome. For intermediate nodes, we only compare the nodes having a path from others. The nodes having no path to each other cannot be compared.

Figure 1 shows a preference graph induced from a CP-Net. The node (*Yes*, *Etiler*, *Low*) represents a low-priced apartment at *Etiler* having a parking area. There is an edge from (*No*, *Kartal*, *High*) to (*No*, *Kartal*, *Medium*). This means that an apartment with a medium price at *Kartal* not having a parking area is preferred over an apartment with a high price at *Kartal* not having a parking area.

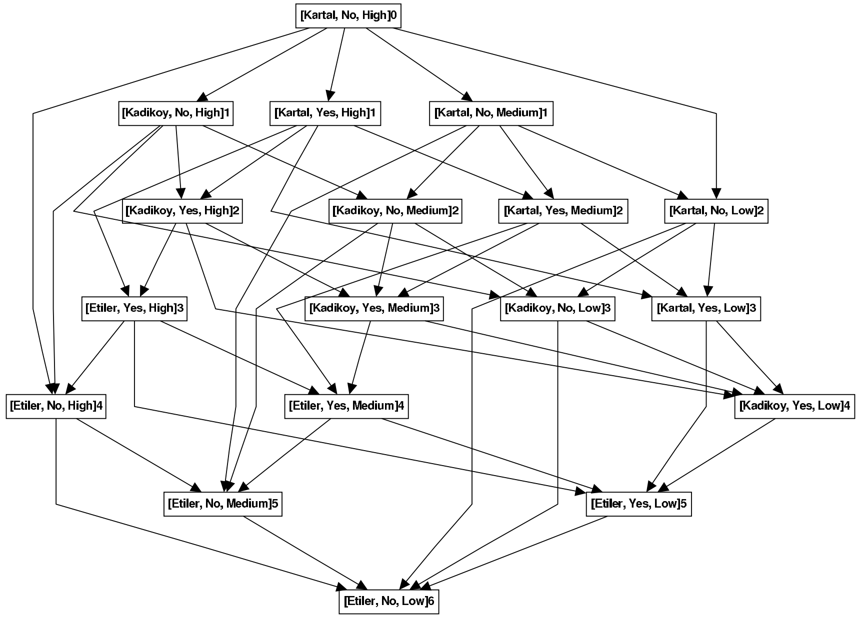

Fig. 1 Induced preference graph from a given CP-Net

An agent having a CP-Net applies one of the following heuristics and uses the estimated utilities produced by a chosen heuristic.

3.1 Depth Heuristic (DH)

We have previously proposed an approach based on capturing the depth of an outcome in preference graph [1] but in that study *depth* is used by the proposed negotiation strategy — it is not independent from the negotiation strategy. However, in this study we use the concept of *depth* to produce estimated utilities of outcomes regardless of negotiation strategy. That is, the agent using this heuristic is able to apply any negotiation strategy.

The depth of an outcome node in a preference graph indicates how far it is from the worst choice. It is intuitive to say that the better (more preferred) a service is, the further it is from the worst outcome. The depth of an outcome node is estimated as the length of the longest path from the root node keeping the worst choice.

According to this approach, the higher the depth of an outcome, the more likely it is to be 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 Equation 2 to estimate the utility values between zero and one. 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 6 in Figure 1, an outcome whose depth is equal to 3 will have utility of $0.5(=3/6)$.

$$U(x) = \frac{Depth(x,PG)}{Depth(PG)} \tag{2}$$

3.2 Borda Scoring Heuristic

CP-Nets order outcomes partially and there are a plenty of linear orderings consistent with the partial ordering of outcomes induced from a CP-Net. One of these linear orderings may reflect the user's real preference orderings. Thus, this heuristic is based on finding all possible linear extensions of a given partial preference ordering and selecting one of the most suitable linear extensions.

One possible way of a linear ordering is to apply a voting procedure. To do this, we estimate all linear extensions of a given partial preference ordering induced from a preference graph and apply a voting procedure called "Borda Rule" [3] to obtain one of the most suitable linear orderings.

According to Borda Rule, we score outcomes according to their position in the ordering. Assume that we have m alternatives ordered as $< o_1, o_2 ... o_m >$ where o_{i+1} is preferred over o_i. When we score the outcomes, each outcome will get a point of its position minus one (o_i will get $i-1$). The sum of points namely *Borda count* represents the aggregation of existing alternative orderings. To illustrate this, consider we have three orderings such as $< x,y,z >$, $< z,x,y >$, $< x,z,y >$ where x, y and z are possible outcomes. Borda count of x would be equal to one $(= 0+1+0)$. In this approach, Borda count of each outcome over all possible linear extensions will reflect how much that outcome is preferred. Thus, we will estimate utilities based on the calculated Borda counts.

On the other hand, the number of all possible linear extensions of a given partial ordering may be so huge that this technique may become impractical because of high complexity. In order to reduce the complexity, we partition the preference graph and apply Borda Rule to all possible linear extensions of each subpartition.

How do we partition the preference graph? We know that the root node holds the worst outcome while the leaf node holds the best outcome. Thus, we need to find an ordering for the outcomes within the intermediate nodes. We partition this part in such a way that each subpartition can involve at most n, predefined number of outcomes. For this purpose, n can be taken as 10 or 15 according to the size of the preference graph. We choose 10 in this study. Figure 2 shows how we partition the preference graph in Figure 1.

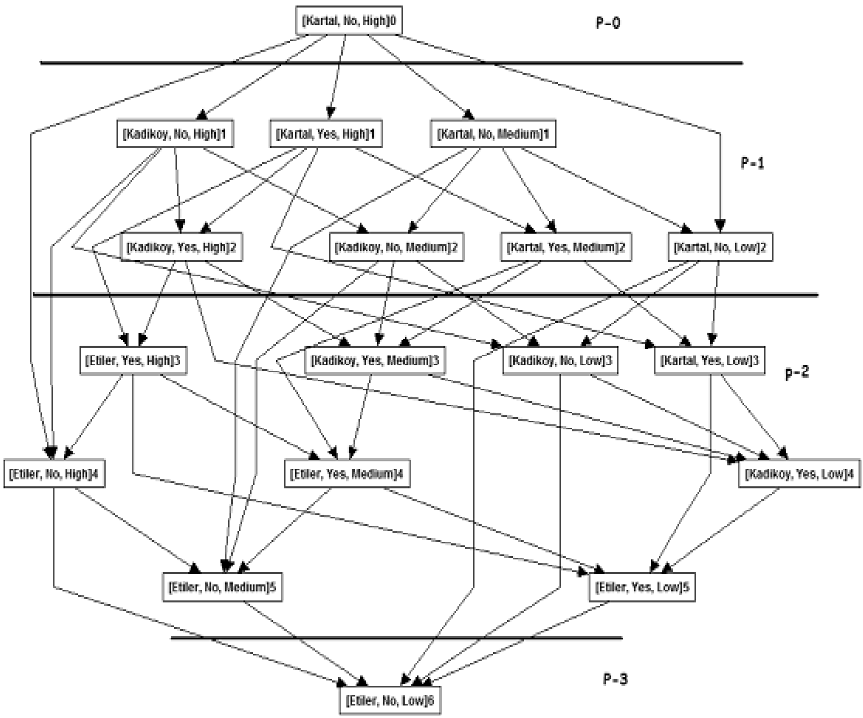

Fig. 2 Partitioning the preference graph in Figure 1

After applying Borda rule to each partition, we normalize Borda counts in a way that Borda count of each outcome will be between zero and one. To do this, we divide Borda count of each outcome in that partition by the maximum Borda count.

Another issue pertains to using these normalized Borda counts in order to estimate final utilities. We distribute the utilities by considering the number of outcomes at each partition. To achieve this, we apply the formula in Equation 3 where $U(x, p_i)$ denotes the utility of outcome x in the i^{th} partition, $U_{max}(i-1)$ denotes the utility of outcome whose utility is maximum in the previous partition $(i-1)$, N denotes the number of possible outcomes, S_{p_i} denotes the number of outcomes in i^{th} partition and $BRCount(x, p_i)$ denotes normalized Borda count of the outcome x. $U_{max}(p_0)$, the utility of worst outcome (root node in the preference graph), is equal to $1/N$.

$$U(x, p_i) = U_{max}(p_{i-1}) + \frac{S_{p_i}}{N} * BRCount(x, p_i) \quad (3)$$

4 Experiments

To evaluate the proposed heuristics, we extend GENIUS [11], which is a platform for bilateral negotiation. Our extension enables an agent to elicit user's preferences as CP-Nets and to use utilities estimated by chosen heuristic while negotiating. The platform also stores the user's total ordering of outcomes as UCP-Nets and evaluates each negotiation outcome for that agent based on the given UCP-Net. The given UCP-Net is consistent with the given CP-net. In our experiments, the UCP-Net serves as ground truth. After an agent negotiates using its CP-Net, we evaluate its performance as if we knew the correct total ordering (UCP-Net).

We investigate three test cases to compare the performance of the heuristics. In each test case, two agents *Agent A* and *Agent B* negotiate with each other. We fix both agents' negotiation strategies so that *Agent A* negotiates with the same *Agent B* (having same preference profile and strategy). In the first case, *Agent A* has a CP-net and applies Depth Heuristic (DH) to derive the estimated utilities. During the negotiation, the agent will act on according to these estimated utilities. In the second case, *Agent A* has the same CP-Net with the first case but it applies Borda Scoring Heuristic (BSH) to estimate utilities which will be used in negotiation. In the last case, *Agent A* has its user's real total preference orderings in the form of UCP-Net (consistent with the CP-net and able to compare all outcomes). Thus, it uses the real utilities. Consequently, we are able to observe what the agent gets at the end of negotiation when it applies heuristics on partial preference information (CP-Net) versus when it has total preference information (UCP-Net).

In our experiments, each agent uses a concession based strategy in which the agent starts with the outcome having the highest utility and concedes over time. It also remembers the best counter offer that is made by the opponent agent. If the utility of the current counter offer is higher than or equal to the utility of agent's previous offer, then the agent will accept the offer. The agent will take the best counter offer of its opponent into account while generating its offer. If the utility of the current offer is lower than that of the best counter offer, the agent will take the opponent's best counter offer.

Since the opponent agent (*Agent B*)'s preference profile has a significant impact on negotiation outcome, we generate 50 different preference profiles for *Agent B*. That is, the same *Agent A* will negotiate with 50 different *Agent B*s. *Agent B*'s preferences are represented with a linear additive utility function in this experiment. Another factor having an influence on negotiation outcome in this setting is UCP-Net of the user. Different UCP-Nets mean different ordering of outcomes, so represent different users. Thus, we generate four different UCP-Nets for *Agent A* consistent with the given CP-net—four different users having the same CP-net. As a result, both agents will negotiate 200 times (4 different users of *Agent A* * 50 different *Agent B*).

Furthermore, we investigate the performance of the heuristics from a different point of view by taking the structure of CP-Nets into account. We generate three different CP-Nets. *CPNet*-1 involves one dependency such as preference of *parking area* depends on *neighborhood* whereas *CPNet*-2 involves two dependencies such as both preferences of *parking area* and *price* depend on *neighborhood*. There are not any dependencies between attributes in *CPNet*-3. For each CP-Net, we generate four different UCP-Nets consistent with them and perform the experiments mentioned above.

4.1 Sum of Utilities for Agent A

Our first evaluation criterion is the sum of negotiation outcomes' utilities with respect to *Agent A* over 50 different negotiations with *Agent B*. Table 1 shows these total utilities for three different CP-Nets and four different UCP-Nets consistent with each CP-Net. As expected *Agent A* using UCP-Net gets the highest score when it has a consistent UCP-Net with *CPNet*-1 and *CPNet*-3 since it negotiates with user's real preference orderings. Overall, the performance of the agent using BSH is quite close to that of the agent using UCP-Net (172 vs. 179 and 171 vs. 172). For the case of *CPNet*-2, the score of BSH is approximately the same as the score of UCP-net. Since *CPNet*-2 involves two dependencies (the user specifies her preferences in a more detailed way), the agent may get more information than the case of other CP-Nets (one dependency and no dependency). This leads to better results. The score of heuristics are the highest when they have *CPNet*-2.

Table 1 Sum of Outcome Utilities over 50 Negotiations for *Agent A*

AGENT A	DH	BSH	UCP-Net
CPNET-1 with UCPNet-1A	39.03	39.00	41.88
CPNET-1 with UCPNet-2A	38.27	40.73	43.66
CPNET-1 with UCPNet-3A	45.73	45.69	45.80
CPNET-1 with UCPNet-4A	46.88	46.94	47.29
Overall Sum (200 negotiations):	**169.91**	**172.36**	**178.63**
CPNET-2 with UCPNet-1B	39.93	41.66	41.70
CPNET-2 with UCPNet-2B	42.94	43.56	43.21
CPNET-2 with UCPNet-3B	46.15	46.76	46.75
CPNET-2 with UCPNet-4B	42.18	43.56	43.53
Overall Sum (200 negotiations):	**171.20**	**175.55**	**175.20**
CPNET-3 with UCPNet-1C	40.17	41.61	40.83
CPNET-3 with UCPNet-2C	41.58	42.50	45.64
CPNET-3 with UCPNet-3C	42.83	43.97	43.37
CPNET-3 with UCPNet-4C	42.36	43.36	42.64
Overall Sum (200 negotiations):	**166.94**	**171.44**	**172.48**

Moreover, *Agent A*'s score while applying Borda Scoring Heuristic (BSH) is higher than the case in which it uses Depth Heuristic (DH) for all CP-Nets (based on overall sum over 200 negotiations). According to this criterion, BSH may be preferred over DH.

4.2 Number of Times as Well as UCP-Net

Our second evaluation criterion is the number of times that the agent that applies a heuristic on a given CP-Net negotiates at least as well as the agent having a UCP-Net. If the utility of outcome for the agent using a heuristic is higher than or equal to the utility of outcome for the agent having UCP-Net, that agent receives one point. Since 50 different *Agent B*s negotiate with the same *Agent A*, we evaluate this criterion over 50 negotiations.

According to Table 2, when *Agent A* uses *CPNet*-1 and applies DH, it negotiates at least as well as the agent having total preference ordering (UCP-Net) in 78 per cent of negotiations whereas BSH is successful at least as UCP-Net in 76 per cent of negotiations. Although the performance of BSH with respect to sum of utilities is better than that of DH, it negotiates as successfully as UCP-Net more than BSH for *CPNet*-1 (78 per cent versus 76 per cent). This stems from the fact that when BSH completes a negotiation better than DH, the difference between utilities of the outcomes is much higher than the case when DH negotiates better than BSH.

Table 2 Number of Times Heuristics Performs As Well As UCP-Nets

AGENT A	DH	BSH
CPNET-1 with UCPNet-1A	40	35
CPNET-1 with UCPNet-2A	26	35
CPNET-1 with UCPNet-3A	46	38
CPNET-1 with UCPNet-4A	44	44
Overall Sum (200 negotiations):	**156**	**152**
CPNET-2 with UCPNet-1B	43	49
CPNET-2 with UCPNet-2B	48	48
CPNET-2 with UCPNet-3B	44	50
CPNET-2 with UCPNet-4B	44	50
Overall Sum (200 negotiations):	**179**	**197**
CPNET-3 with UCPNet-1C	44	50
CPNET-3 with UCPNet-2C	27	31
CPNET-3 with UCPNet-3C	45	49
CPNET-3 with UCPNet-4C	47	47
Overall Sum (200 negotiations):	**163**	**177**

For *CPNet*-2 and *CPNet*-3, the agent using BSH negotiates successfully as the agent having UCP-Net more than the agent using DH. When agents have *CPNet*-2, it is seen that BSH beats DH. Note that in 89.5 per cent of negotiations DH negotiates at least as well as UCP-Net whereas 98.5 per cent of negotiations BSH performs at least as good as the UCP-Net.

5 Discussion

Our experimental results show that it would be better to apply Borda Scoring heuristic (BS) in small domains since its performance is higher than that of Depth heuristic (DH). However, we may prefer to use DH in large domains since its complexity is lower than BSH.

Li *et al.* study the problem of collective decision making with CP-Nets [10]. Their aim 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 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. In contrast, we use *the longest path between the worst outcome and that outcome* to estimate the utilities with our depth heuristic. Moreover, while they propose a procedure for collective decision making, we focus on estimating utility values of each outcome that will be used during the negotiation for an agent.

Rosi *et al.* extend CP-Nets to capture multiple agents' preferences and present *m*CP-Nets [12]. They propose several voting semantics to aggregate agents' qualitative preferences and to determine whether an outcome is preferred over another for those agents. They propose to rank an outcome in term of the length of the shortest sequence of worsening flips between that outcome and one of the optimal outcomes while we use the longest sequence of improving flips between the worst outcome and that outcome in our depth heuristic to get the estimated utilities.

Acknowledgements. This research is supported by Boğaziçi University Research Fund under grant BAP5694 and Turkish Research and Technology Council.

References

1. Aydoğan, R., Yolum, P.: Heuristics for cp-nets: Negotiating effectively with partial preferences. In: HUCOM Workshop in Group Decision and Negotiation, The Netherlands (2010)
2. Baarslag, T., Hindriks, K., Jonker, C., Kraus, S., Lin, R.: 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. SCI, vol. 383, pp. 113–135. Springer, Heidelberg (2012)
3. Black, D.: The Theory of Committees and Elections. Cambridge University Press (1958)
4. Boutilier, C., Bacchus, F., Brafman, R.I.: Ucp-networks: A directed graphical representation of conditional utilities. In: Proceedings of UAI 2001, USA, pp. 56–64 (2001)
5. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. J. Artif. Intell. Res. (JAIR) 21, 135–191 (2004)
6. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. Int. Journal of Robotics and Autonomous Systems 24(3-4), 159–182 (1998)

7. Fatima, S.S., Wooldridge, M., Jennings, N.R.: On Efficient Procedures for Multi-issue Negotiation. In: Fasli, M., Shehory, O. (eds.) TADA/AMEC 2006. LNCS (LNAI), vol. 4452, pp. 31–45. Springer, Heidelberg (2007)
8. Hindriks, K., Tykhonov, D.: Opponent modelling in automated multi-issue negotiation using bayesian learning. In: 7th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS, Estorial, Portugal, pp. 331–338 (2008)
9. Jonker, C.M., Robu, V., Treur, J.: An agent architecture for multi-attribute negotiation using incomplete preference information. JAAMAS 15(2), 221–252 (2007)
10. Li, M., Vo, Q.B., Kowalczyk, R.: An efficient procedure for collective decision-making with cp-nets. In: Proceeding of ECAI 2010, Lisbon, Portugal, pp. 375–380 (2010)
11. Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K., Jonker, C.M.: Genius: An integrated environment for supporting the design of generic automated negotiators. Computational Intelligence (2012)
12. Rossi, F., Venable, K.B., Walsh, T.: mcp nets: Representing and reasoning with preferences of multiple agents. In: Proceeding of AAAI 2004, pp. 729–734 (2004)