A Machine Learning Approach for Mechanism Selection in Complex Negotiations

Aydogan, Reyhan; Marsa Maestre, Ivan; Klein, Mark; Jonker, Catholijn

DOI
10.1007/s11518-018-5369-5

Publication date
2018

Document Version
Final published version

Published in
Journal of Systems Science and Systems Engineering

Citation (APA)
Aydogan, R., Marsa Maestre, I., Klein, M., & Jonker, C. (2018). A Machine Learning Approach for Mechanism Selection in Complex Negotiations. *Journal of Systems Science and Systems Engineering*, *27*(2), 134-155. https://doi.org/10.1007/s11518-018-5369-5

J Syst Sci Syst Eng (Apr 2018) 27(2):134-155
DOI: https://doi.org/10.1007/s11518-018-5369-5
ISSN: 1004-3756 (Paper) 1861-9576 (Online)
CN11-2983/N

A MACHINE LEARNING APPROACH FOR MECHANISM SELECTION IN COMPLEX NEGOTIATIONS

Reyhan Aydoğan[1, 2] Ivan Marsa-Maestre[3] Mark Klein[4] Catholijn M. Jonker[2]

[1] *Department of Computer Science, Özyeğin University, Istanbul, Turkey*
reyhan.aydogan@ozyegin.edu.tr (✉)

[2] *Interactive Intelligence Group, Delft University of Technology, The Netherlands*
C.M.Jonker@tudelft.nl

[3] *Department of Computer Engineering, University of Alcala, 28805 Alcala de Henares, Madrid, Spain*
ivan.marsa@uah.es

[4] *Interactive Center for Collective Intelligence, Massachusetts Institute of Technology, Cambridge, USA*
m_klein@mit.edu

Abstract

Automated negotiation mechanisms can be helpful in contexts where users want to reach mutually satisfactory agreements about issues of shared interest, especially for complex problems with many interdependent issues. A variety of automated negotiation mechanisms have been proposed in the literature. The effectiveness of those mechanisms, however, may depend on the characteristics of the underlying negotiation problem (e.g. on the complexity of participant's utility functions, as well as the degree of conflict between participants). While one mechanism may be a good choice for a negotiation problem, it may be a poor choice for another. In this paper, we pursue the problem of selecting the most effective negotiation mechanism given a particular problem by (1) defining a set of scenario metrics to capture the relevant features of negotiation problems, (2) evaluating the performance of a range of negotiation mechanisms on a diverse test suite of negotiation scenarios, (3) applying machine learning techniques to identify which mechanisms work best with which scenarios, and (4) demonstrating that using these classification rules for mechanism selection enables significantly better negotiation performance than any single mechanism alone.

Keywords: Automated negotiation, mechanism selection, scenario metrics

1. Introduction

Negotiation - the process of finding agreements with self-interested parties with differing preferences - is important in our society, e.g., in commerce, government, but also in science and engineering. In the past few decades, the need to handle increased volumes of transactions, more complex negotiations, and larger number of stakeholders, has driven interest in developing computer-supported negotiation technologies, e.g. tools where software agents facilitate negotiation on behalf of their users (Jennings et al. 2001, Lai and Sycara 2009).

The majority of negotiation research has focused on finding good negotiation mechanisms, which includes a protocol for interactions and strategies for making and accepting offers. The negotiation research community has progressed with respect to negotiation scenarios with a number of independent issues for which the overall negotiation outcome space has an order of magnitude of up to 105 (Aydoğan et al. 2014, Chen 2012, Ammar et al. 2012, Williams et al. 2011, Kraus 2001, Ren and Zhang 2014).

Negotiation mechanisms that work well in those negotiations tend to fare poorly when applied to significantly bigger outcome spaces (Klein et al. 2003). For interdependent issues some effective mechanisms have been proposed (Klein et al. 2003, Ito and Klein 2009, Marsa-Maestre et al. 2012). Still, there are open questions in the field of automated negotiation with respect to what the best mechanisms (protocols and strategies) are, when increasing the number of participants, increasing the outcome space (e.g.1030) and dealing with interdependent issues (which complicates the shape of the negotiation outcome space).

We argue that no single best mechanism exists for all negotiation scenarios, because of the high variability of negotiation scenarios. The variability comes from the variance in the size and shape of negotiation outcome spaces, but also other aspects vary, e.g., whether there is a time pressure, whether other outcome criteria hold (e.g., social welfare and fairness), and to what extent information from the other parties is available. Other complicating factors are whether or not human stakeholders find the mechanism acceptable. For example, a layman user might object to a mechanism because it is not immediately clear that the mechanism has appropriate properties such as guaranteeing a fair outcome.

The central question pursued in this paper is how to select the best negotiation mechanism for a given problem and a set of user requirements. Although this research question has been around for some time (Kersten and Lai 2007), to date the only work within the context of automated negotiation has been done on selecting bidding strategies under the bilateral alternating offer protocol (Ilany and Gal 2016). In comparison, this paper broadens the scope by varying over protocols, and considering interdependent instead of independent issues. This requires a systematic benchmarking method. There have been recent efforts to benchmark negotiation approaches in common scenarios, like the Automated Negotiation Agents Competition (ANAC) (Jonker et al. 2017), but these efforts have restricted to ad-hoc domain sets and, furthermore, the number of nonlinear negotiation scenarios involving interdependent issues is still limited (12 nonlinear scenarios in ANAC 2014).

This paper addresses this challenge and proposes a machine learning approach for mechanism selection in complex negotiations. Our contribution is threefold:

- We create a framework for the characterization and generation of negotiation scenarios (Section 2) by defining a set of scenario metrics to capture the relevant features of negotiation problems (Section 2.1), and by proposing a scenario generation approach which allows us to compile a large and diverse set of negotiation scenarios for mechanism benchmarking in particular nonlinear domains (Section 2.2).

- We evaluate the performance of a range of negotiation mechanisms on our scenario test suite, demonstrating that the relative performance of the mechanisms varies from one setting to another setting (Section 4).
- We build decision trees from our experimental results to map scenario metric values to better-suited protocols, and demonstrate that using these classification rules for mechanism selection enables significantly better negotiation performance than any single mechanism alone (Section 4.2).

The rest of this paper is organized as follows. Section 3 explains our machine learning approach for selecting the best mechanism given the negotiation problem. Finally, Section 5 and Section 6 summarize our contributions, compares our work with other related approaches and discusses future work.

2. Characterization and Generation of Negotiation Scenarios

The first step to fulfill our aim of determining how to select the best negotiation mechanisms for a given negotiation scenario with particular characteristics is to create an infrastructure which allows to systematically benchmark negotiation mechanisms in different scenarios. This infrastructure consists of the following components:

- A set of scenario metrics, which characterize the key properties negotiation scenarios allowing us to divide them in meaningful categories.
- A strategy for scenario generation, which enables us to generate a variety of negotiation scenarios in a systematic way with respect to the specified scenario metrics.
- A set of performance criteria that will be used in benchmarking to evaluate the performance of the mechanisms. Two types of performance criteria are investigated in this study: those related to the negotiation outcome and those related to the negotiation process itself.

2.1 Scenario Metrics

A negotiation scenario is a formal description of a negotiation problem, specifying the following:

- *Negotiation domain* is defined by the set of issues, $ND = \{X_1, ..., X_n\}$ being negotiated over, as well as the valid values for each issues, $\mathrm{Dom}(X_i) = \{x_{i,1}, ..., x_{i,m}\}$. A contract is an assignment of a value $x \in \mathrm{Dom}(X_i)$ to each negotiation issue $X_i \in ND$.
- *Agent preferences* over different contracts in the negotiation domain,

which can be represented in many different ways. For our infrastructure, we adopt the weighted hypercube approach (Ito et al. 2007). According to this approach, a utility function is a collection of hypercube regions in the utility space, each representing a single constraint c_k. A numeric *weight* or *utility value* $u(c_k)$ is associated to each constraint. The utility for a given contract *s* is then calculated as the sum of the utility values for the hypercubes including that contract as shown in equation (1).

$$u(s)=\sum_{c_k \epsilon C \mid Satisfy(s, c_k)} u(c_k) \quad . (1)$$

We have generalized this representation to support "not" constraints, that is, constraints, which are satisfied when the contract ***s*** is *not* contained by the hypervolume. This allows expressing easily utility sinks (i.e. specific regions where utility drops) and "If-then" constraints (i.e. if X holds then Y has to hold too for the constraint to be satisfied). This way to represent preferences has the advantage of being arbitrarily expressive (i.e. given a sufficient number of constraints, we could virtually approximate any imaginable function).

Figure 1 shows a sample utility function for a two-issue negotiation problem. This utility function consists of a unary constraint *C1* and two binary constraint *C2* and *C3*. The corresponding utility values associated to these constraints are 5, 10 and 12 respectively. According to this example, the contract *x* ($issue_1$= 2; $issue_2$=3) would yield a utility value *u(x)*=15 for the agent, since it satisfies both *C1* and *C2* (that is, constraints *C1* and *C2* overlap, creating a region of higher utility). The contract *y* ($issue_1$= 4; $issue_2$=2), on the other hand, would yield a utility value *u(y)*=5, because it only satisfies *C1*.

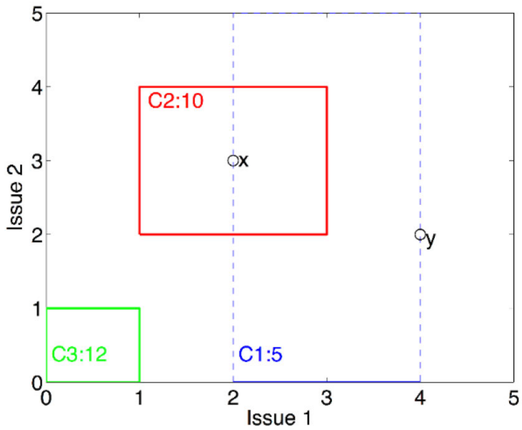

Figure 1. Example of a utility space with two issues and three constraints

Over this definition of a scenario we use a set of metrics for characterization. Some of these metrics have been widely used in the literature. Domain size is measured as the number of possible contracts, and it is directly related to the difficulty to exhaustively search the contract space. Fitness distance correlation (FDC), in contrast, is related to how easy is to exploit the structural properties of the utility functions to find contracts for a given utility value. In particular, it measures the correlation between the utility of a contract and its Euclidean distance, in the contract space, from the global optimum (Tomassini et al. 2005). If there is a strong correlation between distance and utility, this implies a smooth utility function, where it is easy to find utility optima. If the correlation is weak, this implies a rugged utility function with many "bumps" along the way. Other researchers have also used metrics directly related to the constraint-based utility representation, such as average number of constraints, average constraint size, and average constraint dimension (Marsa-Maestre et al. 2012).

Apart from these scenario metrics from the literature, we propose a number of new ones to capture more characteristics of the negotiation problems:

- Statistics of well-known outcome metrics. We sample the scenario utility functions and compute the average and standard deviation for individual utility optimality, social welfare optimality, Pareto optimality and fairness as defined in (Fujita et al 2012). Consequently, we can account for the quality in terms of these different criteria for the expected outcomes if we used random search to find agreements, which serves as a good baseline reference.
- Utility function cross-correlation. We measure the correlation between utility functions for the agents in the same scenario. High cross-correlation means that agent utilities tend to have high values in the same regions, which should allow to find agreements more easily. Low cross-correlation, in contrast, would account for a more "competitive" scenario, where an agent high-gains would imply high losses for other agents.
- Social Welfare FDC. This metric is analogous to FDC above, but instead of being computed over individual agent utilities, we compute it over the social welfare of contracts. In this way, it gives an indication on how rugged is the social welfare landscape of the scenario, which accounts for the difficulty to find areas of high social welfare.
- Attractor metrics. Landscape smoothness is defined taking into account the existence of attraction basins towards its local optima (K., Fogarty, & Miller, 2003). An attraction basin towards a solution *sn* is defined as the set of solutions, *B* (*sn*) which have a continuous trajectory to *sn* where utility never decreases. The size of the basin is given by the cardinality of the set *B* (*sn*). The larger the attraction basins in a landscape are, the smoother the landscape is. We can easily find attractors in a utility space by running hill-climber optimizers from random contracts in the space. If two contracts lead the optimizer to the same local optimum (attractor), we consider them to be in the same attraction basin. We measure both the attractor density of the utility landscape and the average and the standard deviation of the attractor height, that is, the utility values associated with local optima. We compute these metrics for both social welfare and individual agent utilities.

- Veto optimality. A veto hill-climber only progresses when it can make a move which does not decrease the utility of any agent involved in the negotiation (since any agent can veto the move) (Klein et al. 2003). We measure the average social welfare optimality for veto-hill climbers starting from random points. This is an indicator of how easy is to progress towards a negotiation optimum without agents making concessions in their utilities.

With all the metrics outlined above we implemented a set of 22 negotiation scenario metrics shown below:

- Size of contract space
- Average contract utility
- Average local optimum utility
- Agent fitness distance correlation (FDC)
- Average constraint dimension
- Average number of constraints
- Average of natural logarithm of volume of constraints
- Density of utility function optima
- Utility variance of local optima
- Utility variance of random contracts
- Social welfare average
- Pareto average
- Variance of Pareto for random contracts
- Variance of social welfare for random contracts
- Social welfare attractor average height
- Variance of social welfare attractor heights
- Social welfare fitness distance correlation FDC
- Density of social welfare attractors
- Average Fairness: average of all contracts
- Deviation from fairness
- Utility function correlation: for all contracts
- Veto optimality

We have integrated all of those metrics into Negowiki (Marsa-Maestre et al. 2011), an online repository where members of the negotiation research community can upload their scenarios adopting the aforementioned hypercube representation and get them characterized according to the metric set. However, the existing scenario collection, although contained a representative set of scenarios used in the literature, did not cover a wide range of values for the different metrics. Therefore, a systematically created set of diverse negotiation scenarios was needed.

2. 2 Scenario Metrics

We developed, for this work, a scenario generator that uses a parameterized process to define a wide range of utility functions (i.e. sets of weighted hypercubes) for the agents in each negotiation scenario. Hypercubes were generated randomly within the constraints given by the following four parameters:

- Number of issues. The size of the contract space increases exponentially with the number of issues, making it computationally more challenging to find win-win contracts. In our experiments, we generated scenarios with 10, 30, and 50 issues, where each issue had a domain of 10 possible values (0, …, 9).
- Number of shared hypercubes across the agents. If this value is 3, for example, that means that all the agents in the scenario will have 3 hypercubes with the same shape in their utility functions. The similarity between agent utility functions increases when there are many shared hypercubes with equal weights, and decreases when there are many shared hypercubes with opposed weights. In our experiments, every utility function had (2 * number of issues) hypercubes in total, with either no shared hypercubes, 30 % of the hypercubes shared with equal weights, or 30 % of the hypercubes shared with opposed weights.
- Dimensionality distribution for the hypercubes in the utility functions. This specifies what fraction of the hypercubes are uni-dimensional (i.e. constrain the value of just one issue), bi-dimensional (i.e. constrain the values of two issues) and so on. In general, higher-dimensional hypercubes produce more rugged and more difficult-to-optimize utility functions. In our experiments, the scenarios included either only one and two-dimensional hypercubes (equi-probable), or one, two, three and four-dimensional hypercubes.
- Width distribution for the hypercubes in the utility functions. In general, narrow hypercubes create more rugged and difficult-to-optimize utility functions. In our experiments, the scenarios include either narrow hypercubes (with equiprobable widths of one, two, three or four) or wide ones (with equiprobable widths of five, six, seven or eight).

We generated 360 scenarios, consisting of 10 scenarios for each of 36 parameter combinations described above, covering a wide range in terms of the size of the contract space, the similarity of the agent's utility functions, and the ruggedness of the agent's utility functions.

2. 3 Performance Criteria

In order to evaluate the performance of the negotiation mechanisms, we use the following criteria measuring the quality of the outcome from a social-welfare perspective:

- *Utilitarian Social welfare optimality*: Utilitarian social welfare is measured in terms of the sum of the agents' individual utilities (Endriss 2006). Since the value of this metric may vary according to the given scenario, we normalized its value dividing it by the maximum possible social welfare value for the considered scenario.

- *Pareto optimality*: Pareto optimality of an outcome is computed by drawing a line between the zero utility point *z* in the scenario utility diagram and the outcome of the negotiation *o*, and prolonging it until it intersects the Pareto front at a point *p*. We define then Pareto optimality for the contract as the ratio between the length of the segments $\overline{zo}$ and $\overline{zp}$.
- *Fairness*: Outcome fairness is computed as in (Fujita et al 2012), and then normalized to the maximum potential fairness in the considered scenario.

In addition to these outcome performance criteria, we define three process performance criteria:

- *Computation cost*, which accounts for the number of times the utility function is evaluated during the course of the negotiation.
- The *number of rounds* taken by each mechanism to complete the negotiation.
- The *number of messages* exchanged by agents during the negotiation.

All outcome performance criteria are computed by the *Negowiki* when experimental results are uploaded to the website. Process criteria, however, have to be manually uploaded by users, since they strongly depend on the approach implementation.

3. Mechanism Selection Approach

The proposed mechanism selection benefits from machine learning techniques such as decision trees. We take the problem of determining which mechanism would be the best mechanism for a given negotiation problem as a classification problem where the input features are the set of scenario metric values characterizing the given negotiation scenario and the output is the predicted best mechanism for this scenario. In the proposed approach, decision trees are chosen as the classifier method. It is easy for human users to understand the decision trees because of their simple structures. Human experts can deduce important insights by extracting decision rules from decision trees. Furthermore, decision trees have been used for similar purposes. For instance, Guerri and Milano adopt using decision trees to select the best algorithm in an algorithm portfolio in the context of combinatorial auctions (Guerri and Milano 2004). Similarly, Ilany and Gal use decision trees to choose the best negotiation strategy in the context of the alternative offers protocol (Ilany and Gal 2016).

A decision tree is an efficient nonparametric model, which can be used for both regression and classification (Alpaydin 2009). In our case, we use decision trees to classify the best mechanism for a given set of scenario metrics according to a chosen performance criterion. A classification decision tree consists of two types of nodes: leaf node and non-leaf node. The leaf nodes hold the class labels while non-leaf nodes hold the test attributes. Constructing a decision tree for a given problem requires to use training instances. Each instance consists of a set of attributes/features with a class label. Test attributes are chosen from the set of attributes in the given problem. Consider that the given problem involves two attributes: X and Y, and there are three class labels: C1, C2 and C2. Assuming we have the training instances plotted in Figure 2, we may construct the decision tree drawn in Figure 3.

The selection of test attributes is a crucial task in construction phase of the tree since it may affect the size of the tree significantly. These test attributes are used to divide the training instances into subsets by considering the (im)purity of the instances in each group. That is, the quality of the split is measured by the impurity. The split is said to be "pure" if after splitting, all instances in each branch belong to the same class. In the literature, for measuring the impurity, we may use some functions like *entropy* and gini index (Leo Breiman 1984). For all splits, the impurity is calculated - in our work by using *gini index*, and the attribute with the minimum entropy is selected. After selecting a test attribute, tree splits and splitting continues until each subset is homogeneous (i.e., have the same class label) or there is no attribute left for testing. In the following part, we describe how decision trees are used for predicting the best negotiation mechanism for a given negotiation scenario.

Let's examine how we construct the decision trees that will predict the best mechanism for the given negotiation scenario. We follow the following steps:

- We first generate a diverse set of negotiation scenarios and pick a set of different negotiation mechanisms.
- For each scenario, we test every mechanism a number of times and record their performance.
- For each negotiation run, we determine the best mechanism according to the chosen performance criterion.
- We build up the training sets by using the estimated scenario metrics as input features and the name of the best mechanism for each case as the label (i.e., class).
- Then, we create the decision trees whose test attributes are the scenario metrics and leaves are the predicted best mechanism.

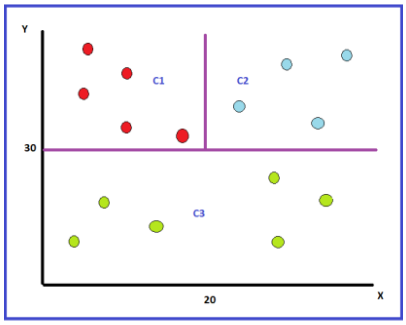

Figure 2. A sample data set with their class information

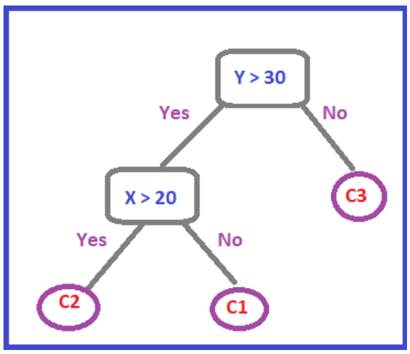

Figure 3. Corresponding decision tree for the dataset depicted in Figure 2

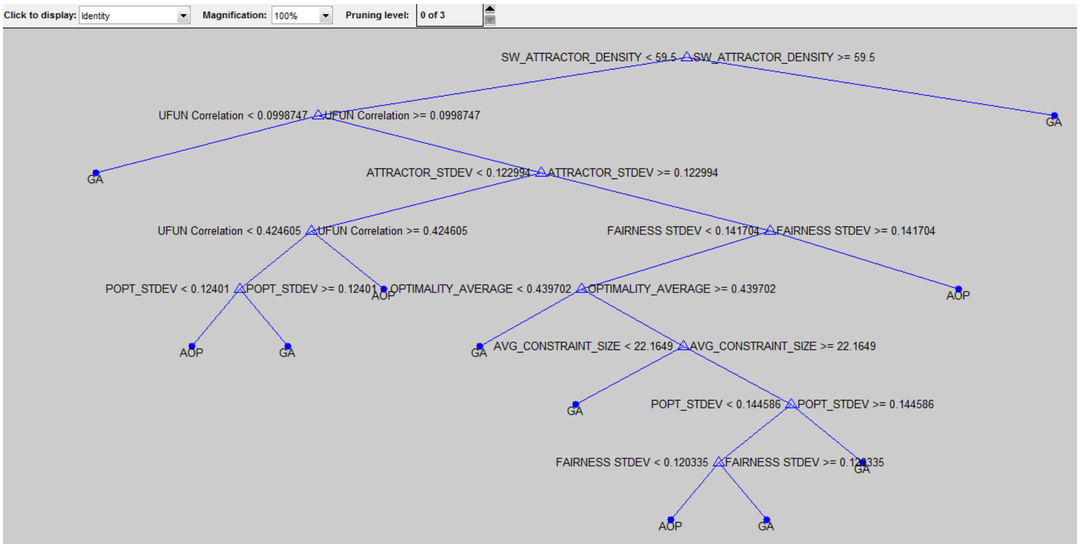

Figure 4. The decision tree for round

Figure 4 depicts the decision tree that has been constructed regarding to the number of rounds criterion. From this decision tree, we can extract some decision rules such as "If social welfare attractor density is less than 59.5 and utility function correlation is less than 0.09999, then it is recommended to use the mediated approach based on Genetic Algorithms (GA)". Note that those rules can be easily interpreted by humans, which is the main reason why we chose to use decision trees.

The decision trees will enable us to predict which mechanism we should pick for the given negotiation scenario. An automated negotiation system can check the prediction of the best mechanism for a given negotiation scenario, and pick the best predicted mechanism in order to gain the best possible outcome regarding the chosen criteria. Note that, such an automated negotiation system can fix the performance criterion such as social welfare and use the

decision tree for that criterion during execution time. Alternatively, it may allow the user to select the performance criterion from a predefined set of criteria dynamically. In this case, decision trees for each performance criterion should be created beforehand and the system picks the one regarding the chosen performance criterion by the user.

Figure 5 depicts the proposed mechanism selection module having the following steps:

- ***Extracting scenario metrics***: We estimate the predefined scenario metrics from a given negotiation scenario. A negotiation scenario consists of negotiating agents' preference profiles (i.e. utility functions) where the scenario metrics capture the characteristics of the underlying scenario (e.g. the complexity of the participants' utility functions as well as the degree of the conflict between participants). As outlined in Section 2.1 for this purposes we have defined 22 scenario metrics as explained in the previous section.

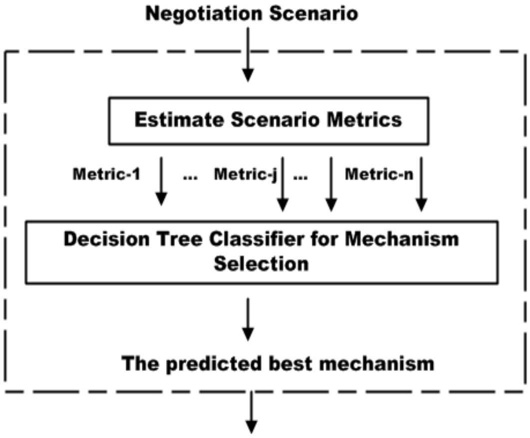

Figure 5. Proposed mechanism selection module

- ***Deciding the best candidate mechanism*:** We use the decision trees to find out the best candidate mechanism for the given negotiation. It is worth noting that for each performance criterion, we constructed a decision tree by using the results of a huge number of negotiations in a variety of negotiation scenarios. These decision trees take the scenario metrics estimated in the former step as inputs and output the best candidate mechanism.

This process requires creating the decision trees in advance. There are a variety of performance criteria for negotiation mechanisms such as fairness, social welfare, cost and so on. One mechanism may be the best mechanism according to fairness whereas another may be the best with respect to social welfare. This leads us to create separate decision trees for each performance criterion.

4. Experiment Settings and Results

The approach we have described in the previous section is fundamentally empirical. Therefore, an extensive experiment set is needed to validate our hypotheses. In the following, we describe our experimental setting and discuss the results obtained.

4. 1 Experiment Settings

As discussed in Section 1, the main hypotheses of our work are that (1) the relative performance of negotiation approaches (i.e. which mechanism works better) varies with the different negotiation settings and performance criteria, and that (2) we can use scenario metrics and machine learning techniques to select which mechanism is more suitable to handle a specific scenario. To put these hypotheses to the test, we have performed three sets of experiments:

- We have performed negotiations over all our generated scenarios using a number of relevant approaches from the literature, and measured their performance according to the criteria discussed above. This has allowed us to check whether there are clear “overall winning” approaches in these complex scenarios.
- We have used our obtained results and the aforementioned scenario metrics to train a

decision tree classifier for the discrimination of the best negotiation approach for each scenario for each of the different performance indicators. We have then used this classifier for mechanism selection, showing that it significantly outperforms any mechanism alone, and also a random selection of mechanisms. This shows how the machine learning approach composed by the metrics and the decision tree is effectively able to recommend negotiation mechanisms for known scenarios.

- We have performed a cross-validation analysis of the decision trees to account for the prediction capability of the decision tree when facing unknown scenarios. The results show how the proposed approach effectively minimizes the performance loss when compared to the optimal performer at each negotiation.

For the negotiations, we created 36 scenario families as described in Section 2.2, varying criteria such as the number of shared constraints across agents, constraint dimensions, constraint with and domain size. For each scenario family we generated 10 instances, for a total of 360 negotiation scenarios.

For each scenario, we ran 10 negotiations with five different negotiation mechanisms taken from the literature:

- Alternating Offers Protocol (AOP). According to the alternating offers protocol (Rubinstein 1982), one of the agents initiates the negotiation with an offer. Other agent can respond by accepting this offer or making a counter-offer or ending negotiation without any consensus. If one of the agent accepts its opponent's offer, the negotiation ends with the agreed offer. Otherwise, this process is iteratively repeated until reaching a deadline (in our experiments, 10000 iterations). In this mechanism, the agents adopt a time-based concession strategy, as described in (Peyman et al. 1998).
- Mediated approach based on Genetic Algorithms (GA). The mediator starts with the generation of N random contracts to propose to the agents, and iteratively each agent selects the top quarter contracts. At each iteration of the mechanism, the mediator uses the agents' selection to recombine and mutate these contracts into a new generation. This is similar to the approach used in (Lin 2004).
- A randomized single text mechanism (veto) (Klein et al. 2003), where the mediator starts at a random contract. At each iteration, the mediator mutates the contract by altering the value of a randomly chosen issue and proposes the new contract to the agents, which will compare it to the previous agreement (i.e. prefer or not prefer). The mediator accepts the proposal as the current agreement if all agents prefer it to the previous contract.
- A simulated annealing single text mediator (SA), which works as veto, but the mediator may accept a non-unanimously agreed contract with a finite probability depending on an annealing temperature which decreases with time. Also, agents are allowed to strongly or weakly accept or reject the current proposed contract at each iteration.

- A pre-negotiation based mechanism (PN). This is a mediated mechanism in which agents first agree about a suitable starting contract. In this pre-negotiation, the mediator proposes a large number of random contracts and the agents perform a runoff voting, i.e., they continue eliminating the candidates with the least votes until just one contract is left (Lang and Fink 2015).

For each negotiation run, we measured computational cost, number of rounds, number of exchanged messages and quality of the outcome (i.e. social welfare optimality, Pareto optimality and fairness). Then we labelled the best performing approach according to each of these six criteria and trained our decision tree classifier for each label set. We used these decision trees as mechanism selection rules and computed the results of doing the negotiations using this mechanism selection approach (MS). Finally, we also tried a random mechanism selection approach (Rand), which basically picked randomly which mechanism to use at each negotiation, to use it as a baseline reference for comparison.

For the cross validation, we created ten disjoint validation sets comprising 10 % of the scenario instances (one of each family), trained the decision trees with the remaining scenarios and tested them against the validation sets, for a total of 3240 negotiations in each training set and 360 negotiations in each validation set.

4. 2 Emprical Results and Analysis

Table 1 shows the results of our experiments with the 3600 negotiation instances. For each negotiation approach, we show the “win count”, that is, the number of times this approach was the best performer according to each criterion. As mentioned above, the MS column corresponds to our mechanism selection approach, where the decision trees are trained using the same 3600 negotiation instances (that is, for *known* scenarios), and the Rand column shows the results using a random mechanism selection. For each row, we highlight in bold the approach with the highest winner count. As seen from the results, adopting the mechanism selected by the decision trees (MS) outperforms sticking to any single protocol for almost all of the performance criteria. Even in the cases when there is an “overall winner”, the results for our MS approach are quite close to that winner, meaning that the decision trees correctly identify the best approach to use most of the times. Furthermore, the results also support the fact that none of the mechanism is the best for all performance criteria. For instance, while alternating offers protocols with conceders (AOP) seems the best performer among the others according to the fairness criterion, MNP is the best performer with respect to the Pareto optimality criterion. Finally, it is worth noting that even when facing known scenarios we don't have a 100 % guarantee of selecting the best performer (there is no cell in the table showing success in the 3600 rounds, not even in the MS column, where we are following the MS recommendation). This is due to the fact that the mechanisms used for complex negotiation scenarios have a high degree of randomness (e.g. performance of SA may highly depend on the first randomly chosen contract). We see, however that following the MS recommendation gives us a significant advantage over any other approach.

In the following, we show the results of the cross-validation experiments. Table 2 shows these results for the computational cost performance indicator. In this case, for each negotiation approach we show the cost differences between the best performer and the given mechanism in each of the 10 validation sets. Therefore, low values in the table means that the approach is closer to an "omniscient" mechanism selector (i.e. one that knew beforehand what was going to happen). We can see that the average difference between the best performer and the mechanism suggested by the decision tree (***MS***) is significantly lower than the distance between the best performer and other mechanisms. It is also worth noting that no mechanisms achieve a zero difference, which means that there is no mechanism which was the best performer in all cases. This again accounts for the fact that in complex negotiation scenarios the degree of variability is much greater. It can also be seen that the different validation sets are consistent with each other. Therefore, for space limitations, we show the results for the rest of the performance indicators averaging all validation sets, as seen in Figure 6.

It is worth noting that there is no overall winner for all performance criteria. For example, in most cases GA outperforms the other mechanisms in terms of the number of rounds to complete the negotiation and number of messages sent during the negotiation while PN outperforms the others in terms of social welfare optimality and Pareto optimality. Although we have such outlier mechanisms, it is shown that the average distance between the best and the mechanism predicted by our selection method is relatively less than others except the winners for the given criterion. This result shows that the decision tree picks the winning mechanism most of the times, and supports our claim of using mechanism selection has advantage over sticking to a particular mechanism.

Table 1 "Win count" for the different approaches in the 3600 scenarios

Criterion	AOP	GA	SA	Veto	PN	MS	Rand
Round	109	3497	0	0	0	**3548**	730
Cost	0	1592	1576	1652	0	**2793**	952
Message	177	3425	0	0	0	**3550**	742
Swopt	141	184	104	395	3089	**3116**	839
Popt	153	261	140	572	**3261**	3152	862
Fairness	1074	674	631	655	678	**1579**	746

Table 2. The cost difference between the best performer and a given protocol in Validation Set

Set	AOP	GA	SA	Veto	PN	MS	Rand
1	1.32 M	61.85	36.99	35.66	8489.78	**14.09**	297797
2	1.77 M	71.95	36.02	33.68	8477.91	**10.33**	387470
3	1.56 M	75.78	31.82	30.04	8488.16	**14.79**	252931
4	1.53 M	62.97	40.04	38.23	8491.99	**12.36**	242828
5	1.42 M	69.58	35.17	33.39	8484.14	**21.99**	231511
6	1.60 M	68.08	37.27	35.91	8484.08	**19.19**	356153
7	1.60 M	62.07	42.04	40.36	8490.10	**17.73**	348657
8	1.33 M	65.66	41.93	40.24	8481.62	**9.59**	286569
9	1.30 M	65.87	38.33	36.79	8489.18	**12.23**	271273

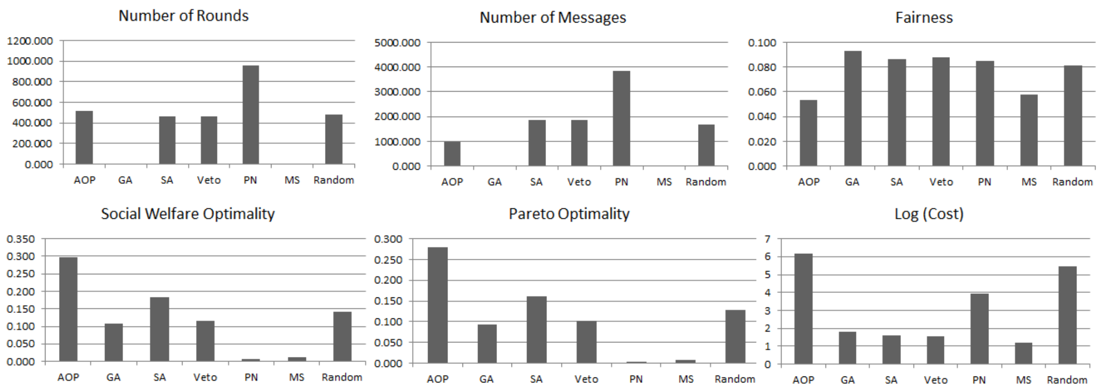

Figure 6. Average distances in validation sets

5. Related Work

In many aspects, the ideas described here have followed the path described by algorithm portfolio design. Algorithm portfolio is a way of determining which algorithm to use (Leyton-Brown et al. 2003). Leyton-Brown *et al.* emphasize that there is rarely a single algorithm, which performs better than all other algorithms for all instances of a given problem. That is, performance of the algorithms may vary depending on the instances of the given problem. In their work (Leyton-Brown et al. 2002), they propose to use the domain knowledge to select the features indicating the distinction of the problem instances and to estimate the running time of the algorithm for each instance of problem. Afterwards, they can use regression to predict the runtime of the algorithm. Following this approach, for a given problem instance, it is possible to calculate the predicted running time for each algorithm and pick the one that performs best.

Ilany and Gal also point out the positive effects of using different algorithms in varying domains in order to achieve better outcomes in the context of automated negotiation (Ilany and Gal, 2016). Their work mainly focuses on selecting a negotiation strategy based on the characteristics of the negotiation scenario. They apply machine learning algorithms to predict which strategy would work better in the given negotiation scenario. The main difference between that study and our work is that they pick the best negotiation strategy guessed by a machine learning algorithm, and employ this strategy in the entire negotiation whereas our approach is aiming to find the best negotiation mechanism, which involves both negotiation protocol and agent strategies, for a given negotiation scenario.

Negotiation has been studied for a couple of decades. Researchers have designed a number of negotiation protocols, which govern the interaction among agents during the negotiation. The alternating offers protocol (Rubinstein, 1982) is one of the most widely used protocols for bilateral negotiation where agents make offers in a turn-taking fashion. Aydoğan *et al.* proposed two extensions of alternative offers protocol for multilateral negotiation (Aydoğan et al. 2017). There are some mediated based multi-party negotiation protocols such as mediated single text negotiation protocol (Klein et al. 2003), feedback and voting based protocol (Aydoğan et al. 2014) in which the mediator makes the offers and negotiating agents vote those offers and give feedback. De Jonge and Sierra introduced a novel multilateral negotiation protocol inspired from human negotiations, called Unstructured Communication Protocol (UCP) (Jonge and Sierra 2015). Bai, Zhang and Sim proposed to use Colored Petri Net models to represent negotiation protocols (Bai et al. 2009). Wong and Fang introduced the Extended ContractNet-Like Multilateral Protocol (ECNPro) (Wong and Fang 2010) for multiple bilateral negotiations between a buyer and multiple sellers. Similarly, William et al proposed a concurrent many bilateral negotiation protocol that allows agents to commit and decommit their agreements (Williams et al. 2012). Sanchez-Anguix *et al.* proposed an extension of alternating offering protocol for team negotiation (Sanchez et al. 2014). The choice of negotiation protocol with compatible negotiation strategies, does not only depend on the dynamics of the negotiation (e.g. the number of negotiating parties, whether it requires a concurrent or single negotiation) but also depends on the specific instance of the negotiation scenario (e.g. competitive versus collaborative). The aim of this work is to provide a machine learning approach to make this decision wisely for bilateral negotiations.

6. Discussion and Conclusion

From the user perspective, it is crucial to be able to map negotiation problems to automated negotiation mechanisms, that is, finding the most appropriate approach to handle a given negotiation problem. However, research works so far have only addressed this challenge in a *per-problem* basis, focusing in specific scenarios and designing negotiation mechanisms suited for those scenarios. Although these works have been successful, the significance of their success cannot be fully assessed, because the results do not provide any insight about how the mechanisms will perform outside of the specific settings they have been tested in.

An approach to allow for consistent comparison of negotiation approaches is benchmarking. A first attempt at it was initiated in 2010, when the first automated negotiating agent competition was organized (Jonker et al. 2017). ANAC allowed to build a common negotiation repository comprising both negotiation scenarios and negotiation strategies for agents, and also provided an infrastructure, the GENIUS testbed (Lin et al. 2014), which allowed for the systematic comparison between strategies in "tournaments". Although it is promising that some researchers have recently started using ANAC repository as a benchmark in order to assess how well their approaches work (Williams et al. 2011, Chen et. al 2012), the ANAC approach has some limitations. First of all, the ANAC scenario repository consists of only scenarios where the negotiating agents' preferences are represented by means of linear additive utility functions, and this scenario repository is generated in a somewhat ad-hoc manner (i.e. competitors submit new scenarios every year), so it is not aimed to provide diversity in a systematic way. Moreover, ANAC agent repository provides agent strategies to be used in a particular negotiation protocol − e.g. Rubinstein's alternating offers protocol (Rubinstein 1982) −, but does not support the possibility to compare complete negotiation mechanisms involving different protocols. Finally, the tournament approach of the ANAC competition supports the "overall winner approach" assumption, which does not help to solve the problem from the user perspective, which is to be able to select which mechanism to use for a given negotiation scenario.

To meet this need, in this paper, we have established a set of scenario metrics based on our previous work in nonlinear negotiation (Marsa-Maestre et al. 2014), and we have generated a wide set of scenarios. We have benchmarked a selection of negotiation approaches from the literature in these settings, and analyzed the results according to performance indicators such as social welfare optimality or negotiation cost in terms of computation. Our results show that no single mechanism is a clear winner for all performance criteria, neither a single mechanism is a winner for all scenarios. The results also show that we can effectively use the aforementioned scenario metrics and a decision tree classifier to map the scenarios onto the most suitable mechanisms to handle them. Finally, we have tested the predictive capabilities of the classification approach, showing that our classifier selects the best performing approach when facing new scenarios in more than 90 % of the cases.

The experiments conducted have validated the hypothesis of this work, and open new avenues for research. We want to explore a wider variety of negotiation scenarios, increasing the number of agents involved in the negotiation and adding the effect of discount factors, for instance. We are also interested in trying different machine learning techniques, such as random forests and extreme learning machines. Finally, we would like to extend our dataset by adding more scenarios and more negotiation mechanisms from the ones available in the literature. We believe that having a large set of scenarios and mechanisms is crucial for the success of any machine-learning mechanism selection approach, and we count on the growing community of *Negowiki* users to achieve that.

7. Acknowledgement

This work was supported by the ITEA M2MGrids Project, grant number ITEA141011, and by the Spanish Ministry of Economy and Competitiveness grants TIN2016-80622-P (AEI/FEDER, UE) and TIN2014-61627-EXP. Many thanks to Mehmet Gönen for his support on machine learning techniques.

References

[1] Aydoğan, R., Festen, D., Hindriks, K. & Jonker, C. M. (2017). Alternating offers protocol for multilateral negotiation. In K. Fujita, Q. Bai, T. Ito, M. Zhang, F. Ren, R. Aydoğan, & R. Hadfi (eds), Modern Approaches to Agent-based Complex Automated Negotiation, pp. 153-167, Springer.

[2] Alpaydin, E. (2009). Introduction to Machine Learning. MIT Press.

[3] Aydoğan, R., Hindriks, K. & Jonker, C. (2014). Multilateral mediated negotiation protocols with feedback. In I. Marsa-Maestre, M. A. Lopez-Carmona, T. Ito, M. Zhang, Q. Bai, & K. Fujita (eds), Novel Insights in Agent based Complex Automated Negotiation, Studies in Computational Intelligence, pp. 43-59, Springer.

[4] Bai, Q., Zhang, M. & Sim, K. M. (2009). Flexible negotiation modelling by using coloured Petri Nets. Journal of Information Technology Research, 2(3): 1-17.

[5] Chen, S., Ammar, H., Tuyls, K. & Weiss, G. (2012). Transfer learning for bilateral multi-issue negotiation, In Proceedings of the 24th Benelux Conference on Artificial Intelligence (BNAIC), pp. 59-66, 2012.

[6] Endriss, U. (2006). Monotonic concession protocols for multilateral negotiation, In Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 392-399, Japan, 2006.

[7] Fujita, K., Ito, T., & Klein, M. (2012). A secure and fair protocol that addresses weaknesses of the Nash bargaining solution in nonlinear negotiation. Group Decision and Negotiation, 21(1): 29-47.

[8] Guerri, A. & Milano, M. (2004). Learning techniques for automatic algorithm portfolio selection, In Proceedings of the 16th European Conference on Artificial Intelligence, pp. 475-479, 2004.

[9] Ilany, L. & Gal, Y. (2016). Algorithm selection in bilateral negotiation. Autonomous Agents and Multi-Agent Systems, 30(4): 697-723.

[10] Ito, T. & Klein, M. (2009). A consensus optimization mechanism among agents based on genetic algorithm for multi-issue negotiation problems, In Proceedings of Joint Agent Workshops and Symposium (JAWS), pp. 286-293, 2009.

[11] Ito, T., Hattori, H. & Klein, M. (2007). Multi-issue negotiation protocol for agents: exploring nonlinear utility spaces. In Proceedings of International Joint Conference on Artificial Intelligence, pp. 1347-1352, 2007.

[12] Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Sierra, C., & Wooldridge, M. (2001). Automated negotiation: prospects, methods and challenges. International Journal of Group Decision and Negotiation, 10(2): 199-215.

[13] Jonge, D. d. & Sierra, C. (2015). Nb3: A multilateral negotiation algorithm for large, nonlinear agreement spaces with limited time. Autonomous Agents and Multi-Agent Systems, 29(5): 896-942.

[14] Jonker, C. M., Aydogan, R., Baarslag, T., Fujita, K., Ito, T. & Hindiks, K. (2017). Automated Negotiating Agents Competition (ANAC), In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), pp. 5070-5072, 2017.

[15] K., V. V., Fogarty, T. C. & Miller, J. F. (2003). Smoothness, ruggedness and neutrality of fitness landscapes: from theory to application. In Advances in Evolutionary Computing: 3-44, Springer.

[16] Kersten, G. E. & Lai, H. (2007). Negotiation support and e-negotiation systems: an overview. Group Decision and Negotiation, 16(6): 553-586.

[17] Klein, M., Faratin, P., Sayama, H. & Bar-Yam, Y. (2003). Protocols for negotiating complex contracts. IEEE Intelligent Systems, 18: 32-38.

[18] Kraus, S. (2001). Strategic Negotiation in Multi-Agent Environments. Cambridge. The MIT Press.

[19] Lai, G. & Sycara, K. (2009). A generic framework for automated multi-attribute negotiation. International Journal of Group Decision and Negotiation, 18(2): 169-187.

[20] Lang, F. & Fink, A. (2015). Learning from the metaheuristics: protocols for automated negotiations. Group Decision and Negotiation, 24(2): 299-332.

[21] Leo Breiman, J. F. (1984). Classification and Regression Trees. Taylor & Francis.

[22] Leyton-Brown, K., Nudelman, E. & Shoham, Y. (2002). Learning the empirical hardness of optimization problems: The case of combinatorial auctions. International Conference on Principles and Practice of Constraint Programming: 556-572, Springer.

[23] Leyton-Brown, K., Nudelman, E., Andrew, G., McFadden, J. & Shoham, Y. (2003). A portfolio approach to algorithm select, In Proceedings of the 18th International Joint Conference on Artificial intelligence, pp. 1542-1543. Morgan Kaufmann Publishers Inc.

[24] Lin, R. (2004). Bilateral multi-issue contract negotiation for task redistribtion using a mediation service, In Proceedings of Agent Mediated Electronic Commerce VI.

[25] Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K. & Jonker, C. M. (2014). Genius: An integrated environment for supporting the design of generic automated negotiators. Computational Intelligence, 30(1): 48-70.

[26] Marsa-Maestre, I., Klein, M., de la Hoz, E. & Lopez-Carmona, M. A. (2011). Negowiki: A set of community tools for the consistent comparison of negotiation approaches, In Proceedings of International Conference on Principles and Practice of Multi-Agent Systems 2011:424-435, Springer.

[27] Marsa-Maestre, I., Klein, M., Jonker, C. M.& Aydogan, R. (2014). From problems to protocols: towards a negotiation handbook. Decision Support Systems, 60(1):39-54.

[28] Marsa-Maestre, I., Lopez-Carmona, M. A., Klein, M., Ito, T.& Fujita, K. (2012). Addressing utility space complexity in negotiations involving highly-uncorrelated, constraint-based utility spaces. Computational Intelligence, 30(1):1-29.

[29] Peyman, F., Sierra, C. & Jennings, N. R. (1998). Negotiation decision functions for autonomous agents. Robotics and Autonomous Systems, 24(3): 159-182.

[30] Ren, F., & Zhang, M. (2014). A single issue negotiation model for agents bargaining in dynamic electronic markets. Decision Support Systems, 60: 55-67.

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

[32] Sanchez-Anguix, V., Aydoğan, R., Julian, V., Garcia-Fornes, A. & Jonker, C. M. (2014). Unanimously acceptable agreements for negotiation teams in unpredictable domains. Electronic Commerce Research and Applications, 13(4): 243-265.

[33] Tomassini, M., Vanneschi, L., Collard, P. & Clergue, M. (2005). A study of fitness distance correlation as a difficulty measure in genetic programming. Evolutionary Computation, 13(2): 213-239.

[34] Williams, C. R., R. V., Gerding, E. H. & Jennings, N. R. (2012). Negotiating concurrently with unknown opponents in complex, real-time domains. Proceedings of 20th European Conference on Artificial Intelligence: 834-839.

[35] Williams, C. R., Robu, V., Gerding, E. H. & Jennings, N. R. (2011). Using gaussian processes to optimise concession in complex negotiations against unknown opponents, In Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp. 432-438, AAAI Press.

[36] Wong, T. & Fang, F. (2010). A multi-agent protocol for multilateral negotiations in supply chain management. International Journal of Production Research, 48(1): 271-299.

Dr. Reyhan Aydoğan is an assistant professor at Özyegin University, Istanbul and guest researcher in Interactive Intelligence Group at Delft University of Technology, the Netherlands. She received her PhD in computer

science from Boğaziçi University in 2011. After then she has worked at Delft University of Technology for 6 years. As a visiting scholar, she has been at Massachusetts Institute of Technology, Norwegian University of Science and Technology and Nagoya Institute of Technology. Her research focuses on the modeling, development and analysis of agent technologies that integrate different aspects of intelligence such as reasoning, decision making and learning. She is applying artificial intelligence techniques such as machine learning and semantic reasoning in designing and developing agent-based decision support systems, particularly negotiation support systems and automated negotiation tools. Dr. Aydoğan is one of the main organizers of the International Automated Negotiating Agents Competition (ANAC). She co-organized the following workshops: Conflict Resolution in Decision Making Workshop (COREDEMA) in PAAMS 2013, ECAI 2016, and IJCAI 2017; The Workshop on Agent-based Complex Automated Negotiations (ACAN) in AAMAS 2015-2017. She is serving as a program committee member in reputable conferences such as AAMAS, IJCAI, and ECAI.

Dr. Ivan Marsa-Maestre is an associate professor at the Computer Engineering Department of the University of Alcala in Spain. He received his telecommunication engineering degree from the University of Alcala in 2003 and the Ph.D. in from the University of Alcala in 2009. His research interests focus on the use of negotiation and nonlinear optimization techniques for distributed coordination of complex systems, such as computer networks, supply chains or vehicle management systems. He has taken part in many public and private research projects in these matters, has a number of publications in high impact international conferences and journals, and serves as program chair and reviewer for some of them. From his research have emerged collaborative research lines with international research groups like the Center for Green Computing, at the Nagoya Institute of Technology (Japan), the Technical University of Delft (TUDelft) or the Center for Collective Intelligence, at the Massachusetts Institute of Technology (USA).

Dr. Mark Klein (http://cci.mit.edu/klein/) is a principal research scientist at the MIT Center for Collective Intelligence. He received his PhD in Artificial Intelligence from the University of Illinois in 1989, and since then has worked for the Hitachi Advanced Research Laboratory, Boeing Research, Pennsylvania State University and (for the last 20 years) the Massachusetts Institute of Technology. He has also had visiting appointments at the Nagoya Institute of Technology, the National Institute of Advanced Industrial Science and Technology, the University of Hong Kong, Otago University, the University of Naples, and the University of Zurich, among others. His research draws from such fields as computer science, economics, operations research, and complexity science to develop and evaluate computer technologies that enable greater 'collective intelligence' in large groups faced with complex decisions. His current projects focus on large-scale on-line deliberation, as well as negotiation protocols for complex problems with many interdependent issues. He has also made contributions in the areas of computer-supported conflict management for collaborative design, design rationale capture, business process re-design, exception handling in workflow and multi-agent systems, and service discovery. He has nearly 200 publications in these areas, with over eight thousand citations on google scholar and an h-index of 43. He serves on the editorial boards of six journals related to AI and social computing, as well as on the program committees for the premier conferences in those areas.

Catholijn Jonker is full professor of Interactive Intelligence (0.8 fte) at the Faculty of Electrical Engineering, Mathematics and Computer Science of the Delft University of Technology and full professor of Explainable Artificial Intelligence (0.2 fte) at the Leiden Institute for Advanced Computer Science of Leiden Universit. She received her PhD in computer science from Utrecht University in 1994. She chaired De Jonge Akademie (Young Academy) of the KNAW (The Royal Netherlands Society of Arts and Sciences) in 2005 and 2006, and she was a member of the same organization from 2005 to 2010. She is a member of the Koninklijke Hollandsche Maarschappij der Wetenschappen and of the Academia Europaea. She was the president of the National Network Female Professors (LNVH) in The Netherlands from September 2013 till January 2016. Catholijn is EurAI Fellow since 2015, and EurAI board member since 2016, EurAI is the European Association for Artificial Intelligence.