Algorithm selection in multilateral negotiation

Luis Fernando Cantú Díaz de León

17 December, 2018

Abstract

Negotiation is an activity that pervades many aspects of human society, and it is bound to become even more important in the future. In a negotiation setting with incomplete information, no single negotiation strategy is optimal for all possible negotiation domains. In this study, we design a meta-agent that addresses this problem by relying on Case-based reasoning and majority voting. The meta-agent is comprised by several other agents, and, in each round of a negotiation, the meta-agent chooses the strategy to follow based on the majority vote of the k agents that were the best performers in similar negotiation domains. These k agents are chosen by means of the k-Nearest Neighbors algorithm. We test our meta-agent by evaluating its average performance against an oracle, the competitors in the 201X Automated Negotiation Agent Competition, and the state-of-the-art agents in the literature.

Keywords: automated negotiation, case-based reasoning, multi-agent systems, majority voting

Contents

Introduction

Negotiation is an activity omnipresent in human societies. It is part of our daily lives and it has been so for many thousands of years [7].

We take part in negotiations in many different social and organizational contexts, all of which are fundamental for the correct functioning of modern societies. Normally, we can find negotiations taking place in activities such as business and trade, but it is also present in other activities, such as political debate in Congress regarding new legislation, or reaching an international agreement over climate change policy. It may play an even more ubiquitous role in a future in which autonomous vehicles negotiate who gets priority in a crossroad, and users haggle with apps in order to obtain more privacy regarding their location, contacts, and photo data [9, 2].

Despite all this constant negotiation taking place all around us, negotiation is not an easy endeavour and we, humans, cannot always reach the best agreements, and sometimes we may even fail to reach one at all. For example, negotiating with others can become a difficult task if their preferences are opposite to our own. In these cases, cooperation or concessions are required in some degree, but not all of us are always willing to engage in those two activities and this may hurt the final outcome of a negotiation [16]. Or, perhaps, one of the parties is unexperienced and cannot handle the complexities of such a situation; it may require additional help in order to succeed.

These difficulties are some of the reasons that have led to the creation of automated negotiators. But creating an automated agent is, as negotiation itself, not an easy task. Many variables must be considered in order to succesfully design an automated agent, including whether its opponent is a human or another automated agent, the number of participants in the negotiation, the agent's own and its opponents' objectives and preferences, the set of issues being negotiated, and the information about the world the agent knows [16, 2].

Nevertheless, automated negotiators have been shown to be useful in many different situations. In the context of human-agent negotiations[1], agents have been able to achieve similar results to, or even outperform, their human counterparts in situations as disimiliar as the *Diplomacy®* game [15], multiple negotiations over a single issue [3], hiring terms between job applicants and employers, and international public health treaties [18].

Agent-agent negotiation has been shown to be useful in several domains, including resource allocation [14], the settlement of energy contracts [4], privacy permission management [1], storage management [20], autonomous driving, internet of things, etc. [2].

Despite the usefulnesss of automated negotiators, a problem remains: many of the agents have been designed to satisfy the chatacteristics of a single domain or a specific opponent, and they, thus, may lack generality and may not yield the outcome intended when facing a different domain or agent [17].

In the case of bilateral negotiation, an *algorithm selection* approach, coupled with machine learning techniques, has been used to design *meta-agents* that generalize better than other agents [10]. This means that, since no particular agent dominates all the others, an agent

[1]Human negotiators, as opposed to their automated counterparts, are subject to bounded rationality, a fact that must influence an agent's design since humans may not necessarily follow the best strategies available nor try to maximize their utility.

should be selected from within an agent library in order for it to tackle a new domain and/or opponent.

The objective of this study is to design an algorithm that provides a solution to the multilateral negotiation version of the *algorithm selection problem*. Instead of using supervised machine learning (SML) techniques, which generalize on observed data, we rely on Case-based reasoning (CBR), which delays generalization until testing time (*lazy learning*). In CBR, a new problem is solved by using or adapting the solutions to similar problems that have been faced before, based on the hypothesis that *similar causes bring about similar effects* [13, 8]. In the context of this study, CBR delivers two different benefits over SML algorithms. Firstly, the fact that CBR is a form of lazy learning allows our meta-agent to change its negotiation strategy over time. Secondly, the data of ongoing negotiation rounds can be stored for future use. Both of these benefits imply that our meta-agent can make better decisions as negotiation rounds go on.

Specifically, we first define a similarity function using the common knowledge of the negotiation setting available to agents. Then we feed this function's output into a *k-Nearest Neighbors* algorithm (kNN), for several values of k, in order to find the agents that perform better in domains which are similar to the new negotiation domain being faced. In every negotiation round, an election takes place among those agents found by kNN[2]. In this election, the agents vote for their preferred action and the action with the most votes is performed. This approach is based on the *wisdom of the crowds* hypothesis: a better solution might be obtained by

[2]The election procedure was first considered by Güneş, Arditi, and Aydoğan [6]. This study generalizes their approach by considering the exclusion of agents specialized in not-so-similar negotiation domains.

aggregating the information and opinions of different individuals [21].

We test our approach by using the negotiation strategies and domains that were submitted to the Automated Negotiation Agent Competition (ANAC)[3]. Our agent was compared with all the participants of the ANAC, with the state-of-the-art algorithms in Ilany and Gal [10] and Güneş, Arditi, and Aydoğan [6], and with an oracle that chooses the optimal strategy available. Three different scenarios were considered:

1. Negotiation domain is in the training set, but not in the test set. Opposing agent is both in the training set and the test set.
2. Negotiation domain is both in the training set and the test set. Opposing agent is in the training set, but not in the test set.
3. Negotiation domain is in the training set, but not in the test set. Opposing agent is in the training set, but not in the test set.

The contributions of this work are manifold. First of all, we define the multilateral negotiation version of the *algorithm selection problem*. Second, we provide the time complexity of this version of the algorithm selection problem. Third, we introduce CBR in the context of automated multilateral negotiations. Fourth, we generalize the approach first considered by Güneş, Arditi, and Aydoğan [6] by excluding some agents, which may lack the necessary level of expertise for a given domain, from voting. Lastly, we design an agent that beats the best performing agent in the ANAC competition and agrees more often with an oracle. The main limitation of this work is that, alhough the strategy proposed may yield good results in other settings, it is only validated in the context of the ANAC competition.

[3]The ANAC takes place every year, we have not yet defined which year's competition we will be using.

This study has the following structure: In Section 1 we discuss the automated multilateral negotiation context, provide the underlying theory regarding agents and their utility maximization problem, and the multilateral negotiation algorithm selection problem are defined and its complexity stated. In Section 2 we discuss the different solutions that have been implemented to design a negotiating agent that is able to generalize and perform well in different negotiation domains. In Section 3, we state the CBR methodology used to provide a solution to the algorithm selection problem. Results are presented in Section 4, and those results are discussed in Section 5.

1 Theoretical framework

1.1 Multilateral negotiation setting[4]

Let $A = \{1, \ldots, n\}$, where $n \in \mathbb{N}$, be the set of agents partaking in a multilateral negotiation. The objective of agents in A is to reach a joint agreement over a set of issues L. L is composed of issues j that can take a value v_j, and $v_j \in V_j$, where V_j is a set of valid values for v_j. In this context, a bid $b^t = (b_1^t, \ldots, b_{|L|}^t)$ consists of an assignment of values to all issues in L at time t. We call the 3-tuple $D = (|L|, \mu_{|V_j|}, |B|)$ the *domain features*, where $|L| \in \mathbb{N}$ is the number of issues in the negotiation, $\mu_{|V_j|} = AVG(|V_j| \; |\forall j \in L)$ is the average number of values in each issue, and $|B| = \prod_{j \in L} |V_j|$ is the number of all possible bids. D is public information known to all agents in A and describes the complexity of the negotiation domain.

[4]The notation and description of a multilateral negotiation are adapted from Ilany and Gal [10] and Güneş, Arditi, and Aydoğan [6].

If all agents in A reach an agreement for a bid b^t at time t, agent i's preferences are represented by the additive linear utility function in Equation (1),

$$u_i(b^t) = \Big[\sum_{j \in L} o_i(b_j^t) w_{i,j}\Big] \delta_i^t \tag{1}$$

Otherwise, its utility is given by $u_i(b^t) = r_i \delta_i^t$, where $o_i(v_j)$ denotes agent i's private valuation of issue j in the given bid, $w_{i,j} \in W_i$ is agent i's weight associated with issue j, δ_i^t is the discount factor at time t, and r_i is agent i's reservation value. $(o_i, W_i, \delta_i^t, r_i)$ is known as agent i's *profile*, which is information privately held by agent i. Several features can be derived from the agent's profile. We call the 5-tuple $P_i = (\delta_i^t, r_i, \sigma_{w_{i,j}}, \mu_{u_i,b^0}, \sigma_{u_i,b^0})$ the *profile features*, where δ_i^t and r_i are defined as above, $\sigma_{w_{i,j}}$ is the standard deviation of weights over all possible issues, μ_{u_i,b^0} is the average utility over all possible bids in the domain at time $t = 0$, and σ_{u_i,b^0} is the standard deviation of agent i's utility over all possible bids in the domain at time $t = 0$. As is the case with the *profile*, the *profile features* P_i is private information and serves as an indication of the complexity of a negotiation from agent i's point of view.

Agents 1 through n engage according to the turn-based *Stacked Alternating Offers Protocol* (SAOP). In SOAP, agents must do one of three different actions $a = \{a_{bid},\ a_{accept},\ a_{end}\}$. This means that agent i can propose a bid to the opposing negotiatiors, accept a bid that has been proposed to him, or walk out of the negotiation. A negotiation ends when any of the following conditions is met:

1. An agreement involving all agents in A is reached.
2. An agent walks out of the negotiation.

3. A deadline is reached.

When an agent i has received a bid b^t from any other agent, then it is able to infer more information. We call the 3-tuple $I_i^t = (u_i(b^t), \mu_{u_i,p^t}, \sigma_{u_i,p^t})$ the *inferred features*, where $u_i(b^t)$ is agent i's utility for bid b at time t, μ_{u_i,p^t} is the average utility of all bids p that agent i prefers over b at time $t = 0$, and σ_{u_i,p^t} is the standard deviation of the utility of all bids p that agent i prefers over b^t at time t. The *inferred features* I_i^t are private information known only to agent i.

1.2 Multilateral negotiation algorithm selection problem

1.3 Complexity of the problem

1.4 Case-based reasoning

In CBR, a solution to a new problem can be found by using or adapting the solutions to similar problems that have been faced before. In order to solve a problem, the retrieval of cases that are similar to the new problem is an important step of the problem solving process [13, 8]. Hence, we need to define what cases are and what similarity is.

Cases represent knowledge tied to specific experienced situations. A case may represent how a task was carried out, how specific knowledge was applied, or which particular strategies were followed to accomplish a goal. A case, when remebered later, might be applicable to a new, but similar, situation [13]. A description of how case bases were built in the multilateral negotiation context can be found in Section 3.

Let X be an arbitrary class of objects. We define the similarity between two objects $x, y \in X$ such that $\sigma : X \times X \mapsto \mathbb{R}_{\geq 0}$. This is also called a *similarity function*. Similarity and distance are strongly related concepts [8]. Let Δ be a related distance, the following are properties required of the similarity function σ:

1. $\forall x, y \in X : \sigma(x, y) \geq 0$ (Non-negativity),
2. $\forall x \in X : \sigma(x, x) = 1$ (Identity of indiscernibles),
3. $\forall x, y \in X : \sigma(x, y) = \sigma(y, x)$ (Symmetry),
4. $\forall x, y, z \in X : \Delta(x, z) \leq \Delta(x, y) + \Delta(y, z)$ (Triangle inequality).

The similarity functions used in the multilareal negotiation context are described in Section 3.

2 Related work

One of the main characteristics of the field of automated negotiation is that many of the agents have been designed to excel at particular negotiation domains, or versus specific kinds of opponents. Although this makes for a better performance in those specific contexts, it also poses a problem: this agents do not perform well when facing a new negotiation domain and/or opponent [17]. This kind of agents, thus, lack generality.

Two different kind of solutions have been proposed to solve this lack of generality problem. The first one involves agents that have been specifically designed to achieve a better performance in more general negotiation contexts, whereas the second one involves an algorithm selection approach in which a negotiation strategy is picked from within a library or database composed of preexisting agents.

The first approach is discussed in Subsection 2.1, whereas the second one is discussed in Subsection 2.2. In Subsection **??** the differences and similarities between our algorithm and those in Subsection 2.2 are stated.

2.1 General agents

Zeng and Sycara [22] introduce a generic agent called *Bazaar* in the bilateral negotiation context. *Bazaar* receives feedback and updates its knowledge after each decision it takes. It models the negotiating environment and the participating agents using Bayesian learning. Faratin, Sierra, and Jennings [5] describe an agent that makes trade-offs in negotiations and seeks to make Pareto-optimal agreements.

In an HP Labs technical report, Karp et al. [12] propose a negotiation strategy based on computer programs that play games such as chess The authors treat bilateral negotiation as a two-player game in which a game tree that results from enforcing the negotiation protocol is created. For every offer, the agent looks at every counteroffer, every counteroffer of this counteroffer, and son on. The agent traverses the tree by selecting the counteroffer with the largest expected payoff.

An agent that uses its opponents history of bids to predict its preferences is described in Jonker, Robu, and Treur [11]. The agent can use its opponents revealed preferences, although incomplete, to improve the efficiency of the agreements. They validate their approach for a total of 96 negotiation traces and conclude that their agent is able to achieve agreements that lie relatively close to the Pareto-efficient frontier.

2.2 Generalizing by solving the algorithm selection problem

Ilany and Gal [10] introduced the algorithm selection problem in the bilateral negotiation context. In their study, they analyze two different versions of this problem: off-line and on-line.

In the off-line version, strategies cannot be changed over time. The problem is solved with the help of a *meta-agent*, which makes use of supervised machine learning algorithms trained in order to discern which of the agents within a pool of agents performs better under which scenarios. The structural characteristics of the negotiation domain, the past performance of ANAC 2012 agents, and other agent's privately known features are used as input to train the machine leraning models. If the meta-agent is facing a known domain, then the best performing agent in that domain is selected. Every time the meta-agent faces a new domain, a prediction is made and a negotiation strategy is selected according to the results of this prediction. The authors compare the performance of this meta-agent with that of the best average agent of the ANAC 2012 and an oracle that selects the optimal negotiation strategy retrospectively. The comparison is made under three different scenarios: new domain, known agent; known domain, new agent; and new domain, new agent. They find that the meta-agent performs better than the best average agent in all of those scenarios, but the meta-agent only agreed with the oracle 40% of the time, and it does not perform well in some scenarios because it is unable to change its strategy after it has decided on it.

To solve the on-line version of the algorithm selection problem, a Multi-Armed Bandit (MAB) approach was used by the authors. Specifically, they use the Upper Confidence Bound (UCB) algorithm, which provides an upper bound of regret for not choosing the optimal "arm" all of the time, to model the exploitation-exploration tradeoff intrinsical to the MAB problem. Two different versions of this algorithm are presented. The first one is a pure MAB approach that makes no prior assumptions about the performance of the strategies at its disposal. The second one does include these prior assumptions by incorporating the insights of the off-line algorithm described before.

Finally, they show that the MAB approach with the prior assumptions has a better performance than the meta-agent and the pure-MAB agent in one setting, and performs better than the competing agents in ANAC 2013.

Güneş, Arditi, and Aydoğan [6] employs a different strategy to design a meta-agent that generalizes well in multilateral negotiation settings. Although the authors draw inspiration from the algorithm selection problem, they bring other ideas into the mix, such as *genetic algorithms* and *mixture of experts*. Two different, but related, algorithms are proposed.

1. The *incremental portfolio* algorithm: If the meta-agent is the first to offer, then it chooses the bid that gets it the highest utility. If the meta-agent receives an offer from an opponent, a referendum takes place in which the agents within the meta-agent vote over two different options: to accept the offer or to make a bid. If the accept option wins the referendum, then the meta-agent accepts its opponent's offer. Otherwise, every agent proposes a bid, and one of them is selected randomly. Experts' bids could be weighted according to their expertise.

2. The *crossover* algorithm: If the meta-agent is the first to offer, then it chooses the bid that gets it the highest utility (same as incremental portfolio). If the meta-agent receives an offer from an opponent, a referendum takes place in which the agents within the meta-agent vote over two different options: to accept the offer or to make a bid. If the accept option wins the referendum, then the meta-agent accepts its opponent's offer. Otherwise, agents vote over the value of each issue. The value with the majority of votes wins and is picked as the bid. In this case, experts' votes could be weighted according to their expertise.

The authors compare both of their meta-agents, with both unequally and equally weighted agents, with the top five average agents in the ANAC 2016 competition. Both versions of the *crossover* algorithm perform better than their opponents, whereas the versions of the *incremental portfolio* algorithm rank rather lowly.

2.3 Comparison

Table 1 presents the similarities and differences to be found between our own algorithm and those presented in Ilany and Gal [10] and Güneş, Arditi, and Aydoğan [6].

Table 1: Similarities and differences between our algorithm and those in related works.

		[10]		[6]	
Concept	Our algorithm	Meta-agent	Prior MAB	Incremental portfolio	Crossover
Rational agents	1	1	1	1	1
Bilateral negotiations	0	1	1	0	0
Multilateral negotiations	1	0	0	1	1
Agents allowed to model their opponents	0	0	0	0	0
Agents can switch strategies during negotiations	1	0	1	1	1
Similarity function	1	0	0	0	0
Case-based reasoning solution	1	0	0	0	0
Voting or Referendum	1	0	0	1	1
Machine Learning algorithms	possibly	1	1	0	0
Multi-Armed Bandit solution	0	0	1	0	0

3 Methodology

In this section, we describe the methodology used. In subsection 3.1, we show the different case bases that were proposed. In subsection ??, we discuss the different functions used to calculate the similarity between different cases. In subsection 3.3, an algorithm that solves the algorithm selection problem is proposed. Finally, in subsection 3.4, we describe the experiments that were carried out in order to validate our algorithm.

3.1 Case bases[5]

Agent i's view of a negotiation can be seen as a sequence of the tuples D, P_i, I_i^t, which were defined in subsection 1.1. The values of the tuple I_i^t change with each passing t, whereas the values of D and P_i remain constant during a specific negotiation. The grouping of this sequence constitutes a time series. Several different approaches were tried to build the case base:

- *Single:* there is a case for each t in D, P_i, I_i^t. This means that we have a time window of size 1, which implies that the cases are not time series. The main advantage of this approach is that the meta-agent is able to make use of its case base from $t = 1$ and onwards.
- *Consecutive:* there is a case every n bids. We have a case for $t = 1, \dots, n$, another one for $t = n + 1, \dots, 2n$, and so on. This means that we have a time window of size n, which implies that cases are time series of length n. The main advantage is that we can

[5]The building of the different case bases was based on Lora Ariza, Sánchez-Ruiz, and González-Calero [19].

exploit the time series structure, whereas the main drawback is that the meta-agent can only use its case base every n bids.

- *Overlapped:* describes the evolution of the values of D, P_i, I_i^t during the last n bids. We have a case for $t = 1, \ldots, n$, another one for $t = 2, \ldots, n+1$, and so on. The cases are time series of length n. The main advantage is that we can exploit the time series structure, whereas the main drawback is that the meta-agent can only use its case base after the first n bids.
- *From start:* there is a new case for every t, describing the evolution of the values of D, P_i, I_i^t from the beginning of the negotiation to the current t. In this approach, each time series has a different length.

3.2 Similarity functions

Let $\mathbb{D}$ be the set of all training cases d, and let $\mathbb{D}'$ be the set of all test cases d'. Let A be the set of training agents, and let A' be the set of test agents.

Let $\sigma : \mathbb{D} \times \mathbb{D}' \mapsto \mathbb{R}_{\geq 0}$ be the similarity function. The similarity function used was the Euclidean distance.[6] Specifically, the similarity between two cases was computed as a linear combination of the similarities between their time series.

[6] In the experimentation phase, this might change in order to include Machine Learning algorithms or other kind of similirity functions.

3.3 Algorithm

The following algorithm provides a solution for the algorithm selection problem:

- If the meta-agent is the first to move then it chooses the bid, i.e. a_{bid}, that gets it the highest utility.
- Otherwise, σ is used to find the set of k-Nearest Domain Neighbors $kNDN = \{d_1, \ldots, d_k\} \in \mathbb{D}$ of the new instance $d' \in \mathbb{D}'$. Only the k agents that gained the highest average utility in (d')'s nearest neighbors in $kNDN$ are allowed to vote.
  - If the meta-agent receives an offer from an opponent, a referendum takes place in which the k agents within the meta-agent vote over two different options: a_{accept} and a_{bid}. If we have a_{accept} for a number of agents greater than $k/2$, then meta-agent i accepts its opposing party's bid. Otherwise, agents cast a vote $\forall v_j \in V_j$. If some v_j obtains the majority of votes, then v_j is picked as the bid b^t in period t.

3.4 Validation

We test our approach by using the negotiation strategies and domains that were submitted to the Automated Negotiation Agent Competition (ANAC)[7]. Our agent was compared with all the participants of the ANAC, with the state-of-the-art algorithms in Ilany and Gal [10] and Güneş, Arditi, and Aydoğan [6], and with an oracle that chooses the optimal strategy available. The agent that obtains the highest average individual utility is considered the best performing agent.

[7] The ANAC takes place every year, we have not yet defined which year's competition we will be using.

Three different scenarios are considered:

1. Negotiation domain is in the training set, but not in the test set. Opposing agent is both in the training set and the test set.
2. Negotiation domain is both in the training set and the test set. Opposing agent is in the training set, but not in the test set.
3. Negotiation domain is in the training set, but not in the test set. Opposing agent is in the training set, but not in the test set.

4 Results

5 Discussion and conclusion

References

[1] Tim Baarslag et al. "An Automated Negotiation Agent for Permission Management". In: *Proceedings of the 2017 International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '17. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 380–390. URL: http://dl.acm.org/citation.cfm?id=3091125.3091184.

[2] Tim Baarslag et al. "When will negotiation agents be able to represent us? The challenges and opportunities for autonomous negotiators". In: *Proceedings of the Twenty-sixth International Joint Conference on Artificial Intelligence*. IJCAI'17. Melbourne,

Australia, 2017, pp. 4684–4690. DOI: 10.24963/ijcai.2017/653. URL: https://doi.org/10.24963/ijcai.2017/653.

[3] Andrew Byde et al. "AutONA: A system for automated multiple 1-1 negotiation". In: *E-Commerce, 2003. CEC 2003. IEEE International Conference on*. IEEE. 2003, pp. 59–67.

[4] Shantanu Chakraborty, Tim Baarslag, and Michael Kaisers. "Energy Contract Settlements through Automated Negotiation in Residential Cooperatives". In: *2018 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE. 2018. URL: https://arxiv.org/abs/1807.10978.

[5] P. Faratin, C. Sierra, and N.R. Jennings. "Using similarity criteria to make issue trade-offs in automated negotiations". In: *Artificial Intelligence* 142.2 (2002). International Conference on MultiAgent Systems 2000, pp. 205–237. ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(02)00290-4. URL: http://www.sciencedirect.com/science/article/pii/S0004370202002904.

[6] Taha D. Güneş, Emir Arditi, and Reyhan Aydoğan. "Collective Voice of Experts in Multilateral Negotiation". In: *PRIMA 2017: Principles and Practice of Multi-Agent Systems*. Ed. by Bo An et al. Cham: Springer International Publishing, 2017, pp. 450–458. ISBN: 978-3-319-69131-2.

[7] Yuval Noah Harari. *Sapiens: A Brief History of Humankind*. HarperCollins, 2015. ISBN: 9780062316103.

[8] Eyke Hüllermeier. *Case-Based Approximate Reasoning (Theory and Decision Library B)*. Berlin, Heidelberg: Springer-Verlag, 2007. ISBN: 140205694X.

[9] Matthew Hutson. *How artificial intelligence could negotiate better deals for humans*. Science Magazine article. 2017. URL: https://www.sciencemag.org/news/2017/09/how-artificial-intelligence-could-negotiate-better-deals-humans.

[10] Litan Ilany and Ya'akov Gal. "Algorithm selection in bilateral negotiation". In: *Autonomous Agents and Multi-Agent Systems* 30.4 (2016), pp. 697–723.

[11] Catholijn M. Jonker, Valentin Robu, and Jan Treur. "An agent architecture for multi-attribute negotiation using incomplete preference information". In: *Autonomous Agents and Multi-Agent Systems* 15.2 (Oct. 2007), pp. 221–252. ISSN: 1573-7454. DOI: 10.1007/s10458-006-9009-y. URL: https://doi.org/10.1007/s10458-006-9009-y.

[12] Alan H. Karp et al. "A Game Tree Strategy for Automated Negotiation". In: *Proceedings of the 5th ACM Conference on Electronic Commerce*. EC '04. New York, NY, USA: ACM, 2004, pp. 228–229. ISBN: 1-58113-771-0. DOI: 10.1145/988772.988807. URL: http://doi.acm.org/10.1145/988772.988807.

[13] Janet Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.

[14] Sarit Kraus. *Strategic negotiation in multiagent environments*. MIT Press, 2001.

[15] Sarit Kraus and Daniel Lehmann. "Designing and building a negotiating automated agent". In: *Computational Intelligence* 11.1 (1995), pp. 132–171.

[16] Raz Lin and Sarit Kraus. "Can Automated Agents Proficiently Negotiate with Humans?" In: *Communications of the ACM* 53.1 (Jan. 2010), pp. 78–88. ISSN: 0001-0782. DOI: 10.1145/1629175.1629199. URL: http://doi.acm.org/10.1145/1629175.1629199.

[17] Raz Lin et al. “Genius: An integrated environment for supporting the design of generic automated negotiators”. In: *Computational Intelligence* 30.1 (2014), pp. 48–70. DOI: 10.1111/j.1467-8640.2012.00463.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8640.2012.00463.x.

[18] Raz Lin et al. “Negotiating with bounded rational agents in environments with incomplete information using an automated agent”. In: *Artificial Intelligence* 172.6 (2008), pp. 823–851. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2007.09.007. URL: http://www.sciencedirect.com/science/article/pii/S0004370207001518.

[19] Diana Sofía Lora Ariza, Antonio A. Sánchez-Ruiz, and Pedro A. González-Calero. “Time Series and Case-Based Reasoning for an Intelligent Tetris Game”. In: *Case-Based Reasoning Research and Development*. Ed. by David W. Aha and Jean Lieber. Cham: Springer International Publishing, 2017, pp. 185–199. ISBN: 978-3-319-61030-6.

[20] Shay Raz, Raz Lin, and Onn Shehory. “Collaborative Load-Balancing in Storage Networks Using Agent Negotiation”. In: *Cooperative Information Agents XII*. Ed. by Matthias Klusch, Michal Pěchouček, and Axel Polleres. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 306–320. ISBN: 978-3-540-85834-8.

[21] James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[22] Dajun Zeng and Katia Sycara. “Bayesian learning in negotiation”. English (US). In: *International Journal of Human Computer Studies* 48.1 (Jan. 1998), pp. 125–141. ISSN: 1071-5819.